ALIAS: Analysing Legal Implications and Agent Information Systems, number IR-CS-004

Brazier, F.M.; Oskamp, A.; Prins, J.E.J.; Schellekens, M.H.M.; Schreuders, E.; Wijngaards, N.J.E.; Apistola, M.; Voulon, M.B.; Kubbe, O.

2003

document version
Publisher's PDF, also known as Version of record

ALIAS

Analysing Legal Implications and Agent Information Systems

prof.dr. F.M.T. Brazier[1]
prof.dr.mr. A. Oskamp[2]
prof.dr.mr. J.E.J. Prins[3]
dr.mr.ir. M.H.M. Schellekens[3]
dr.mr. E. Schreuders[3]
dr. N.J.E. Wijngaards[1]
ing. M. Apistola[2]
mr. M.B. Voulon[2]
O. Kubbe[1]

July 2003

http://www.iids.org/alias

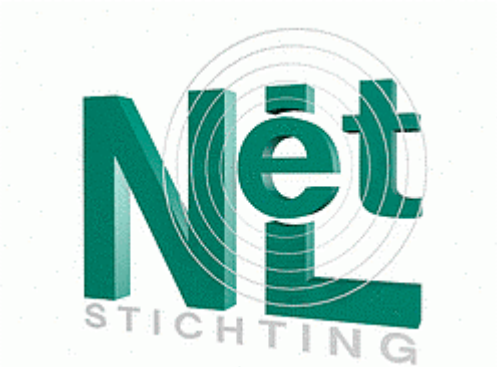
[1] Intelligent Interactive Distributed Systems, Faculty of Sciences
Vrije Universiteit Amsterdam, de Boelelaan 1081a, 1081 HV, Amsterdam, The Netherlands
Email: {f.m.t.brazier, n.j.e.wijngaards}@cs.vu.nl
Phone: +31 - 20 - 444 7737, 7756; Fax: +31 - 20 - 444 7653

[2] Computer and Law Institute, Faculty of Law
Vrije Universiteit Amsterdam, de Boelelaan 1105, 1081 HV, Amsterdam, The Netherlands
Email: {a.oskamp}@rechten.vu.nl
Phone: +31 - 20 - 444 6215; Fax: +31 - 20 - 444 6230

[3] Center for Law, Public Administration and Informatization, Faculty of Law
University of Tilburg, P.O. Box 90153, 5000 LE, Tilburg, The Netherlands
Email: {J.E.J.Prins, M.H.M.Schellekens}@uvt.nl
Phone: +31 - 13 - 466 3088, 8044; Fax: +31 - 13 - 466 8149

Technical Report no. IR-CS-004,
Faculty of Sciences,
Vrije Universiteit Amsterdam.

The ALIAS project is a multi-disciplinary project specifically aimed at exploring the legal status of agents and the implications of their use, `http://www.iids.org/alias/`.



The ALIAS project is supported by NLnet Foundation, `http://www.nlnet.nl/`.

CONTENTS

1 INTRODUCTION 1

1.1 ALIAS PROJECT 1
1.1.1 Research Goal 1
1.1.2 Research Partners 2
1.1.3 Research Methodology 2
1.1.4 Research Results 2

1.2 SOFTWARE AGENTS 3
1.2.1 Agents from an AI Perspective 3
1.2.2 Agents from a CS Perspective 4
1.2.3 Software Agents: New Challenges for Law 5
1.2.4 Agents from a Legal Perspective 5

1.3 LAW AND LEGAL ASPECTS 6
1.3.1 Some Basics 6
1.3.2 Civil law vs. Common Law 6

1.4 A CONCEPTUAL MODEL FOR LAW AND COMPUTER SCIENCE 7

1.5 STRUCTURE OF THIS DOCUMENT 7

2 AUTONOMY 9

2.1 ILLUSTRATIVE SCENARIOS 9
2.1.1 Pre-Contractual Phase 9
2.1.2 Closure of Contracts 9
2.1.3 Liability for Faults in the (Data) Processing by Agents 10
2.1.4 Support of Organisational Processes 10

2.2 RECIPE FOR THE PRE-CONTRACTUAL PHASE 10
2.2.1 Legal Analysis about Pre-Contractual Liability 11
2.2.1.1 Legal requirements during the pre-contractual phase 11
2.2.1.2 Relevant parties and possible measures 12
2.2.1.3 Legal Requirements 14
2.2.2 Technical Considerations 15

2.3 RECIPE FOR THE CLOSURE OF CONTRACTS 16
2.3.1 Legal Considerations 16
2.3.1.1 Legal background 16
2.3.1.2 Information exchange during contract formation 17
2.3.1.3 Considerations and recommendations for the designer of software agents 19
2.3.2 Agent Specific Technical Measures to Close Contracts. 22
2.3.2.1 Technical measures for the producer of the software agent: 22
2.3.2.2 Technical measures for the producer of the software agent and the supplier of goods: 22
2.3.2.3 The producer of the software agent: 22
2.3.2.4 The consumer 22
2.3.2.5 The producer of the agent 22
2.3.2.6 The agent platform 23

2.4 RECIPE CONCERNING LIABILITY FOR FAULTS IN THE (DATA) PROCESSING BY AGENTS 23
2.4.1 Legal Analysis 23
2.4.1.1 Who has to take precautionary measures? 24
2.4.1.2 To what length does one have to go in taking precautionary measures? 24
2.4.2 Technical Measures to Prevent Liability 26
2.4.2.1 Measures to reduce liability for the designer or programmer of the software 27
2.4.2.2 Measures to prevent liability for the user of the agent 27
2.4.2.3 Measures to prevent liability for the owner of an agent 27
2.4.2.4 Measures to prevent liability for the person making the agent available to the user 28
2.4.2.5 Measures to prevent liability for the agent platform supporting agents 28

2.5 RECIPE FOR THE SUPPORT OF ORGANISATIONAL PROCESSES 28
2.5.1 Legal Analysis 29
2.5.1.1 Labour law aspects of planning by software agents 29
2.5.1.2 Legal aspects of knowledge management by software agents 30
2.5.1.3 Legal Requirements 31
2.5.2 Technical Considerations 32
2.5.2.1 Planning 32
2.5.2.2 Knowledge management 32
2.6 DISCUSSION 33

3. IDENTIFIABILITY AND TRACEABILITY 35
3.1 ILLUSTRATING SCENARIO 35
3.1.1 Identification 35
3.1.2 Anonymity 35
3.1.3 Facilitation of Anonymity 35
3.2 RECIPE FOR SELF-IDENTIFICATION 36
3.2.1 Legal Analysis 36
3.2.1.1 Identification duties 36
3.2.1.2 Identification duties during contract formation 38
3.2.1.3 Legal requirements 40
3.2.2 Technical Perspective 40
3.2.2.1 Agent identities and human identities 40
3.2.2.2 Techniques for identification 41
3.2.2.3 Measures for identity protection 43
3.3 RECIPE FOR ANONYMITY 43
3.3.1 Legal Analysis 44
3.3.1.1 Definition issues 44
3.3.1.2 Legal status of anonymity 44
3.3.1.3 Limitations of anonymity 44
3.3.1.4 Legal protection of anonymity 45
3.3.1.5 Towards a right to anonymity? 46
3.3.1.6 Legal requirements 46
3.3.2 Technical Analysis 46
3.3.2.1 User anonymity 47
3.3.2.2 Agent anonymity 47
3.3.2.3 Agent platform anonymity 47
3.4 FACILITATING ANONYMITY AND PSEUDONYMITY 48
3.4.1 Legal Aspects 48
3.4.1.1 When is it unlawful to offer services that allow others to act anonymously or using a pseudonym? 48
3.4.1.2 When must providers of services that allow others to act anonymously or using a pseudonym, provide the means to remove the anonymity and pseudonymity of these others? 49
3.4.1.3 Should persons acting anonymously or using a pseudonym be notified that their true identity can be traced? 49
3.4.1.4 Legal requirements 49
3.4.2 Technical Analysis 50
3.4.2.1 Authorisation and access control 50
3.4.2.2 Risks & recovery 50
3.4.2.3 Services 51
3.5 DISCUSSION 51

4 INTEGRITY & ORIGINALITY 53
4.1 ILLUSTRATING SCENARIO 53
4.1.1 Integrity 53
4.1.2 Integrity of Evidentiary Data 53
4.1.3 Originality 53

4.2 INTEGRITY 53
4.2.1 Legal Analysis 53
4.2.1.1 Criminal law protection of integrity 54
4.2.1.2 The bearer of the data 55
4.2.1.3 Privacy and telecommunications law 56
4.2.1.4 Legal requirements 57
4.2.2 Technical Perspective 58
4.2.2.1 Data integrity 58
4.2.2.2 Agent integrity 58
4.2.2.3 Agent platform integrity 59
4.3 INTEGRITY OF EVIDENTIARY DATA 60
4.3.1 Legal Analysis 60
4.3.1.1 Law of evidence 61
4.3.1.2 Legal requirements promoting the production and integrity of evidentiary data 61
4.3.1.3 Evaluation 62
4.3.1.4 Legal requirements 62
4.3.2 Technical Perspective 63
4.3.2.1 Evidentiary data 63
4.3.2.2 Integrity 64
4.3.2.3 Role of parties involved 64
4.4 ORIGINALITY 64
4.4.1 Legal Analysis 64
4.4.1.1 The evidentiary function 65
4.4.1.2 The 'uniqueness' function of originality 65
4.4.1.3 Legal requirements 67
4.4.2 Technical Perspective 68
4.5 DISCUSSION 68

5 TRUST 71
5.1 ILLUSTRATING SCENARIO 71
5.1.1 Reliability 72
5.1.2 Confidentiality 72
5.1.3 Non-Excessiveness 72
5.2 RECIPE FOR RELIABILITY 72
5.2.1 Legal Analysis 72
5.2.1.1 Communication 72
5.2.1.2 Legal requirements 73
5.2.2 Technical Analysis 74
5.2.2.1 Reliable communication 74
5.2.2.2 Reliable processing 75
5.3 RECIPE FOR CONFIDENTIALITY 75
5.3.1 Legal Analysis 75
5.3.1.1 Definitions 75
5.3.1.2 Interference with computer systems 76
5.3.1.3 Computer intrusion 76
5.3.1.4 Interception of data transmissions 76
5.3.1.5 Destruction of computer data 76
5.3.1.6 Holders of confidential information 77
5.3.1.7 Legal requirements 77
5.3.2 Technical Aspects 77
5.4 RECIPE FOR NON-EXCESSIVENESS 78
5.4.1 Legal Analysis 78
5.4.1.1 Collection of personal data 79
5.4.1.2 Processing of personal data 79
5.4.1.3 Termination of processing 79
5.4.1.4 Legal requirements 79
5.4.2 Technical Analysis 79
5.4.2.1 Non-excessiveness for agent platforms 80
5.4.2.2 Non-excessiveness for agents 80

6 DISCUSSION 81
6.1 OPEN AND CLOSED SYSTEMS 81
6.2 BALANCING INTERESTS 82
6.2.1 Tension 1 - Sharing and Shielding Information 82
6.2.2 Tension 2 - Control and Dependency 83
6.2.3 Tension 3 - Freedom and Obligations 83
6.3 CONCLUSIONS 84

REFERENCES 85

A SCENARIOS 93
A.1 INTRODUCTION 93
A.1.1 Grocery Shopping 93
A.1.2 Chemical Commodities Market 93
A.1.3 Hospital 93
A.1.4 Local Government 93
A.1.5 Open & Closed Systems 93
A.1.5.1 Open systems 94
A.1.5.1 Closed Systems 94
A.2 GROCERY SHOPPING 94
A.2.1 Context 94
A.2.2 Agents 95
A.2.3 Example 95
A.2.3.1 Situation 1: human grocery shopping 95
A.2.3.2 Situation 2: shopping with personal assistant 95
A.2.3.3 Issues with grocery shopping 96
A.2 CHEMICAL COMMODITIES MARKET 96
A.2.1 Context 96
A.2.2 Agents 97
A.2.3 Examples of Chemical Marketplaces 97
A.2.3.1 Situation1: The 'blackboard' marketplace 97
A.2.3.2 Situation 2: The Auction hall 97
A.2.3.3 Situation 3: Match making 97
A.2.3.4 Situation 4: Automated negotiation 97
A.2.3.5 Situation 5: Transportation 98
A.2.3.6 Issues with agents in the marketplace 98
A.3 THE HOSPITAL 98
A.3.1 Context 98
A.3.1.1 The medical division 99
A.3.1.2 The nursing division 99
A.3.1.3 Knowledge management 99
A.3.1.4 Information Technology (IT) support 100
A.3.1.5 Problems 100
A.3.1.6 Wishes 100
A.3.1.7 Solutions 101
A.3.1.8 Current research 101
A.3.2 Agents 101
A.4 LOCAL GOVERNMENT 102
A.4.1 Context 102
A.4.2 Agents 102
A.4.3 Example 102
A.4.3.1 Situation 1: Just in time information management 102
A.4.3.2 Situation 2: Labour resource assistance 103
A.4.3.3 Issues with labour resource management and information management. 103

B LEGAL FRAMEWORK 105
B.1 WHAT ROLES OF AGENTS ARE TO BE CHOSEN? 105
The agent as a data processor 105
The agent as a processor of personal data 106
The agent as a means of communication 106
The agent as a provider of services 106
The agent as a means to conclude contracts 106
The agent as a product to be consumed 107
B.2 CONCLUDING REMARKS 107

1 INTRODUCTION

The global use of the Internet provides an enormous incentive for the general public, governments, and companies to explore the new possibilities of electronic information exchange. Electronic environments originate, with new characteristics, not only technically but also socially and legally. E-commerce is a typical example of a new phenomenon with many different types of implications. Local government automation of front-office functions is another example. The provision of passports to citizens through Internet, automated traffic fines and taxation are recent examples of changes due to new technology.

Technology on the Internet involves multiple co-operating processes (e.g., known as services, peers, or agents). Software agents are the most versatile of these processes, as their characteristics such as autonomy, mobility, and intelligence are characteristics taken from human interaction. Notions such as anonymity and privacy need to be reconsidered in the 'digital world'. In the ALIAS project the research areas of Computer Science, Artificial Intelligence, and Law are combined to analyse legal implications (rights, responsibilities, problems and new questions) of agent technology. The aim of this project is to provide guidelines for both CS/AI-researchers and Legal-researchers.

This document describes the exploratory results from the ALIAS project, without claiming exhaustiveness. A number of problems are identified (e.g., in relation to open-ended systems, free for all to join, and closed systems with controlled access) and directions for future research are indicated.

Chapter 1 is structured as follows. Section 1.1 describes the goals and structure of the ALIAS project. Section 1.2 defines the concept of a software agent. Section 1.3 describes the role of the law and legal aspects in relation to software agents. Section 1.4 introduces a conceptual model for Law and Computer Science, which forms the basis for the structure of this document, as described in Section 1.5.

1.1 ALIAS PROJECT

The notion of an agent often includes several properties such as autonomy, mobility and intelligence, and tasks such as negotiation, co-operation, learning, etc. (see Section 1.2 for a more detailed description of software agents). Properties and tasks are developed by man. The way people interact in the real world increasingly finds its reflection in the 'digital world'. Notions such as responsibility, identity and anonymity acquire new meanings. New concepts such as pseudo anonymity emerge. Agents can be deployed for several purposes, all with their own social and legal consequences. Until now, much research on deployment of information technology has been done within separate disciplines. The ALIAS-project is an interdisciplinary explorative project designed to study the extent to which legal issues can map onto technical concepts and conditions. Section 1.1.1 describes ALIAS research goal; Section 1.1.2 the research partners, Section 1.1.3 the research methodology and Section 1.1.4 research results.

1.1.1 Research Goal

The ALIAS project (`http://www.iids.org/alias`) is an interdisciplinary explorative project that studies the legal and technical implications of the use of software agents. This research is partly motivated by the assumption that social acceptability of agent technology will increase if the legal context is more clearly identified. Also the assumption that interdisciplinary research can be of help in drawing a more realistic picture of the possible applications of agent technology is a motivation of this research.

Computer Science and AI develop the technical expertise and applications. Law fits these applications into existing legal and regulatory frameworks, or creates new frameworks. In this project the disciplines CS, AI and Law aim to collaborate in an earlier stage. The legal perspective takes into account both the Anglican (US and UK) and continental European legal traditions. In addition to relevant national rules from both legal traditions, the legal analyses deal with existing sources of international law (such as United Nations treaties and legislative acts from the European Union, Anglican Agents perform tasks. Often, these tasks have legal consequences: e.g., a law may pose rights and obligations upon the owners or users of agents when it comes to the responsibility for the agent's acts. Under certain conditions some of these rights and obligations may be transferred to other parties by means of a contract. The technical conditions needed to allow legal acts to be performed for and by agents is one of the subjects of this research project. The result is that concepts such as anonymity, pseudo-anonymity, traceability and liability for distributed multi-agent systems are viewed from a *new* perspective.

To recapitulate the research goal:

> To investigate the legal and consequently some of the social implications of agent technology, which should ultimately lead to recommendations for the development of distributed multi-agent systems as well as recommendations for changes in the existing legal and regulatory framework.

The explorative project started February 2001, and will finish in 2003, after which ALIAS successor projects will commence. This explorative ALIAS project is funded by Stichting NLnet (http://www.nlnet.nl). NLnet's goal is to foster Internet research and development. The ALIAS project is to provide a 'cookbook' for (1) designers of agent systems on the way to design (and code) intelligent agents in a legally valid way and (2) legal experts, policy makers as well as other relevant parties indicating where new laws or clarification is needed with respect to existing law. The general assumption in this document is that (at least initially) the agent domain is to be constrained to so-called 'rational' agents. Agents in contexts as evolutionary computing, neurocomputing, and simulation are examples of research that are not (yet) addressed.

1.1.2 Research Partners

Three institutes have worked together in this project. The VU Computer Law Institute at the Vrije Universiteit in Amsterdam studies both the legal implications of the use of IT and the prospects and limits of using IT for legal practice. Agents are one of the key research issues. The Intelligent Interactive Distributed Systems Group at the same university focuses on scalable distributed multi-agent systems. The Center for Law, Public Administration and Informatization of the University of Tilburg focuses its research on legal implications of Information and Communication Technologies, regulatory issues concerning ICT and re-conceptualisation of law in light of developments such as de-materialization, de-territorialisation, de-identification and loss of human involvement.

1.1.3 Research Methodology

An iterative research cycle is used within the ALIAS project, fuelled by descriptions of scenarios situated in the real world. The basis for this research is formed and limited to three key research areas: Agent technology, Law and Computer Systems. Scenarios are used to acquire insight in the broadness of the field: e.g., the types of tasks agents can/may perform, the situations in which they are used, the roles they can play. Four scenarios are included in this document: grocery shopping, chemical commodities marketplace, local government automation and hospital automation, on the basis of which similarities and differences were analysed. Relevant legal and technical implications are analysed, e.g., resulting in the distinction between open and closed systems in which agents may operate. Intermediary concepts are developed to overcome language and conceptual barriers between legal experts and computer scientists, forming the basis for a structured analysis of legal aspects of software agents.

1.1.4 Research Results

The ALIAS project has disseminated its results in a number of ways: papers, posters, workshop and conference participation (including the LEA workshops and the WOGLI workshop), and this technical report. The workshops demonstrate that this field of research is gaining momentum, and will play an important role in the next decade, as electronic agents (or their equivalents) are more commonly used.

In the year 2001, Oskamp and Brazier (2001) presented the paper "Intelligent agents for lawyers" at the Workshop on Legal Knowledge Systems in Action: Practical AI in Today's Law Offices.

In the year 2002, Brazier and Oskamp gave an invited talk at the 17th British & Irish Law, Education and Technology Association (BILETA'02) conference titled "Agents: Nomads, Migrants or Globetrotters?" (2002), and presented the paper (Apistola, Brazier, Kubbe, Oskamp, Schellekens and Voulon, 2002a) regarding legal aspects of agent technology, which was also presented at the Belgian-Dutch Conference on Artificial Intelligence (BNAIC'02) (2002b). The paper by Brazier, Kubbe, Oskamp and Wijngaards (2002) regarding "Are Law-Abiding Agents Realistic?" was presented at the Law and Electronic Agents (LEA'02) workshop. A poster on "Migrating Agents: do Sysadmins have a License to Kill?" was presented at the System Administration and Networking (SANE'02) conference (Apistola, Brazier, Kubbe, Oskamp, Prins, Schellekens, and Voulon, 2002).

In the year 2003, three papers were presented at the Law and Electronic Agents (LEA'03) workshop (Brazier, Oskamp, Prins, Schellekens and Wijngaards, 2003; Brazier, Oskamp, Schellekens and Wijngaards, 2003a; 2003b), which were based on material from Chapters 2, 3 and 4. Brazier and Oskamp also gave a presentation at the Working Group on the Law and Intelligent Information Agents (WOGLI'03) workshop during the Agentcities meeting in Barcelona.

The aforementioned 'cookbook' refers to the document currently being read: an amalgamation of the explorative research of this project.

1.2 SOFTWARE AGENTS

In this endeavour to research legal aspects of software agents it is of interest to investigate what is actually meant by terms such as 'agent', 'multi-agent', 'distributed multi-agent', not only in a technical sense but also in the context of law. The aim is not to give a conclusive answer on to what an agent exactly is, will be or should be. The hope is to create a clearer picture to foster a frame of reference.

Agents are used in a wide variety of applications (Jennings and Wooldridge, 1998), including process control, manufacturing, air traffic control, information management, electronic commerce, business process management, patient monitoring, health care, games, and interactive theater and cinema. The artificial intelligence (AI) community, the Computer Systems (CS) community and the legal community all research agents and agent systems, each from their own perspective. This section describes current research on agents and agent systems from all three perspectives.

1.2.1 Agents from an AI Perspective

The AI perspective on agents is related to research in Distributed Artificial Intelligence and Computer Supported Co-operative Work. In essence, distributed activities are conceptually modelled by application of the agent metaphor (e.g., Jennings, 2000), which encompasses both human and automated agents. An agent, human or automated, has its own environment, which consists of other agents and a (material) world. The main characteristics of agents, from the AI perspective (e.g., Bradshaw, 1997; Nwana, 1996; Wooldridge and Jennings, 1995; Shoham, 1993) are listed below:

- Agents are autonomous.
- Agents may be mobile (i.e., are capable of migrating to different locations on the Internet).
- Agents are capable of communication with other agents and interaction with their environment.
- Agents are most often intelligent (i.e., are capable of learning, have knowledge, can perform complex tasks, can reason about and with this knowledge).

Autonomy is an important characteristic of agents: when they roam the Internet they are 'on their own', with limited guidance. Jennings and Wooldridge (1998) define autonomous behaviour as: '… the system should be able to act without the direct intervention of humans (or other agents) and should have control over its own actions and internal state'. This definition is augmented by Castelfranchi (1998; 2000) by distinguishing two kinds of autonomy: autonomy in delegation and autonomy as independence. An agent is completely autonomous (relative to a given goal) when the agent does not need the help or resources of other agents (including humans) to achieve its goal.

A number of (formally specified) agent models have been developed, some of which have a degree of genericity. These agent models structure the agent's internal processes based on an underlying framework, e.g., the generic agent model (Brazier, Jonker and Treur, 2000) and agent models for BDI-agents (e.g., Rao and Georgeff, 1995; Meyer and Schobbens, 1999).

Agent communication is actively researched, with a focus on agent communication languages (ACLs), ontologies, and co-operation models, resulting in many different languages and models, leading to interoperability problems (Wooldridge and Jennings, 1999). An ACL defines message exchanges, and is often based on a formal model based on 'speech acts', e.g., KQML (Finin, Labrou and Mayfield, 1997). FIPA (Dale and Mamdani, 2001) is involved in standardisation of ACLs and protocols (as co-ordination models) (FIPA, 2000). The semantics of an ACL are generally left open to be instantiated with an ontology. An ontology (Gruber, 1993) is an explicit specification of a conceptualisation, i.e., a domain-specific definition of concepts, objects, and other entities, plus relations among those. Usually, ontologies are formally specified, and are used to provide semantics to information. Many different ontologies have been developed, again leading to problems of a Babel nature.

The Semantic Web effort (Berners-Lee, Hendler and Lassila, 2001; Ding, Fensel, Klein and Omelayenko, 2002) offers a solution, as it aims to develop ontologies with which machines can understand information. Other approaches involve intermediary agents, capable of translating between ontologies, or even capable of acting as intermediaries or gateways (Maes, 1994; Levy, Sagiv and Srivastava, 1994; Sycara and Zeng, 1996). Co-ordination models may be used to overcome interoperability problems by replacing message exchange, and setting up and maintaining aggregates of co-operating agents. An ensemble, consisting of a number of integrated agents, executes as a whole. This implies that temporal and referential coupling between agents in message-

based systems is replaced by an associative memory that decouples communicating parties in time and space (Omicini and Papadopoulos, 2001).

An agent's behaviour can often not be optimised in advance, as its environment is unpredictable: the behaviour of other agents is unknown, and the environment may change in unexpected ways. Agents have to learn from and adapt to their environment, i.e., the ability of agents to learn how to co-operate and to compete (Alonso, D'Inverno, Kudenko, Luck and Noble, 2001). Profiling, i.e., learning and maintaining information about (human) agents, is one of the current applications of adaptation and learning techniques, mostly applied in negotiation settings (e.g., Bui, Kieronska and Venkatesh, 1996; Dastani, Jacobs, Jonker and Treur, 2001). In stead of changing an agent's beliefs about its environment, an agent's code can also be adapted (Brazier, Overeinder, Steen and Wijngaards, 2002).

The global behaviour of a multi-agent system is difficult to predict as it is an emergent property of the interactions between different, autonomous, agents. Current research focuses on self-organisation and self-regulation based on emergent properties (Holland, 1995). Broadly speaking, two approaches are used: small agents, and big agents. The small agents approach is inspired by biology (e.g., ants), in that many small agents with individual simplistic behaviour yield surprisingly complex behaviour (e.g., Bonabeau and Theraulaz, 2000). The big agents approach is based on heterogeneous agents in a dynamic environment, and is currently researched to a lesser extent because of the complexity in formally describing aggregated behaviour of rational agents.

In sum, agents are actively studied within AI and are used to perform tasks. The behaviour of individual agents and multi-agent systems is an important topic of current research. Implementation of agents has not received much attention within AI, except for demonstration purposes.

1.2.2 Agents from a CS Perspective

The CS perspective on agents and multi-agent systems is related to distributed systems and middleware research. An agent is often considered to be just a process (Tanenbaum and Steen, 2002): a piece of running code with data and state, although functionality of agents can be described in terms of human behaviour, sometimes warranting the predicate 'intelligent'. Agents are processes that are autonomous and pro-active (capable of making "their own" decisions when they like), interacting, and may be mobile.

Agents, services and objects are often confused, as all are entities which contain data and can be interacted with. Objects, however, are generally considered to be passive (Jennings and Wooldridge, 1998), while services are active (and possibly autonomous), while agents are autonomous, active, and mobile:

- An object needs to be invoked in order to perform a function, and performs only during an invocation.
- A service receives messages and autonomously decides if, when, and how to (re)act; the agent may also perform functions even if not requested to do so.
- An agent has in addition to a service's abilities the ability to migrate.

Agents interact with each other and with services, usually by exchanging messages and interact with objects by invoking functions on objects. Message exchange and function invocation are subject to different 'qualities of service'. For example, the message paradigm described by FIPA prescribes reliable and ordered point-to-point communication between agents (Fipa, 2001), and remote function invocation may involve blocking (i.e., waiting for the object's function to finish) or non-blocking invocations (Tanenbaum and Steen, 2002).

Mobility of agents involves mobility of processes, although the agent autonomously decides to migrate. Process mobility means that the process can be migrated to a location with required resources or services, which entails. migrating the process's code and data (e.g., Fugetta, Picco and Vigna, 1998). The main reasons for using mobile code is to overcome the lack of resources to run processes locally, and the desire to share resources and improve load balancing among distributed systems (e.g. see Lange and Oshima, 1999). An example application is information retrieval on the web (e.g., Gudivada, Raghavan, Grosky and Kasanagottu, 1997; Odubiyi, Kocur, Weinstein, Wakim, Srivastava, Gokey and Graham, 1997) in which agents migrate to the information collections, in stead of remotely analysing information.

Agents cannot exist in a vacuum, and need support, commonly provided by an *agent platform* for:

- creating and running an agent,
- searching for an agent,
- migrating agents to other platforms,
- enabling communication with other agents hosted by agent platforms.

Most of the agent platforms employ the Java Virtual Machine (JVM), which provides object serialization as a basic mechanism to implement weak mobility. The JVM does not provide mechanisms to deal with the

execution state for strong mobility. A number of agent platforms support strong migration of agents, including NOMADS (Suri, Bradshaw, Breedy, Groth, Hill, Jeffers, Mitrovich, Pouliot and Smith, 2000), Ara (Peine and Stolpmann, 1997), and D'Agents (Gray, Cybenko, Kotz, Peterson and Rus, 2002). Most agent platforms, however, support weak mobility, including Ajanta (Tripathi, Karnik, Vora, Ahmed and Singh, 1999), and Aglets (Lange, Oshima, Karjoth and Kosaka, 1996). Some agent platforms do not provide mobility, e.g. OAA (Martin, Cheyer and Moran, 1999) and RETSINA (Sycara, Paolucci, Velsen and Giampapa, 2001). Aspects such as interoperability, efficiency and performance, but also security, are part of current research on agent platforms, e.g., in AgentScape (Wijngaards, Overeinder, Steen and Brazier, 2002) or standardisation efforts such as FIPA (http://www.fipa.org) and OMG/MASIF (Milojicic, Breugst, Busse, Campbell, Covaci, Friedman, Kosaka, Lange, Ono, Oshima, Tham, Virdhagriswaran and White, 1998).

Agent systems need to be secure, as mobile agents can easily become the next generation of viruses. Current research on secure agent systems concentrates mainly on protecting hosts against hostile mobile agents: unknown code is run on a local machine, with possible devastating effects. The reverse, protecting mobile agents from hostile hosts, is realised by a few number of frameworks, including Ajanta (Karnik and Tripathi, 2001) and AgentScape (Noordeinde, Brazier and Tanenbaum, 2002).

1.2.3 Software Agents: New Challenges for Law

Although some ten years ago authors argued there is no room for law in cyberspace and that the virtual world cannot and should not be controlled by rules and regulations, the present state of affairs proves them wrong. All over the world, the amount of formal legislation (for example, statutes and treaties) and private arrangements (such as codes of conduct or contracts) aiming to restrict the power of large *e*-companies and protecting the free flow of information on the Internet, the privacy of its users, and the copyright of titleholders is steadily growing. In addition, legislative projects are introduced in order to provide the actors in the virtual world with certainty as regards the applicability of various well-known rules established long ago to be applied in a world of bricks and mortar. Law does affect behaviour, outside as well as within the virtual world. However, the application of existing law does not always lead to desirable outcomes. For example, the present scope of copyright protection tends to threaten the fundamental freedom of information. Furthermore, serious problems arise with regard to the predictability and enforceability of the law. Traditional concepts, rules, and principles, based on the physical world, do not completely fit the new reality.

It is obvious that the introduction of agent technology is again an example that ICT has important consequences for our legal system. World-wide references can be found in legal literature to the fact that traditional legal instruments are inadequate for regulating the various consequences of the introduction of agent technology. These consequences can be divided into questions on the one hand and challenges on the other. In the case of questions, the focus is generally on problems to which a concrete answer can be provided via regulations: for example 'Can an agent perform a legal act?' In the case of challenges, the focus is not so much on the formulation of a single concrete answer to a — usually — practical question. Instead the problem is more often associated with finding a balance or weighing up between various — often contradictory — interests (for example the weighing up of the interest of knowing the identity of the owner of an agent on the one hand and the interest of anonymity in acting by means of an agent on the other, when deciding whether and under what conditions agents may be used). In principle, a number of answers are possible to the questions and challenges, while it is not always clear whether these are also definitive answers. As becomes clear in this report, legal questions as well as legal challenges arise in relation to the design and use of agent technology. This implies that sometimes specific suggestions can be given as regards the design of agents, whereas in other situations one single answer or suggestion for designing agents cannot be given.

1.2.4 Agents from a Legal Perspective

From a legal perspective an agent is 'just' a process, i.e., a program in action, which looks after certain interests of its user in a networked on-line environment. This environment is characterized by the presence of other users who have their own set of interests and possibly their own set of agents. An agent may take care of its user's interests with varying degrees of autonomy and intelligent behaviour. For the purpose of this brief legal analysis, three roles of agents are distinguished, depending on the degree of autonomy with which they operate:

- A *slave* has no autonomy at all: it needs to consult its user ('master') for any decision that affects its user's possessions, legal rights and obligations.
- A *representative* may take its own decisions within a well-defined domain and within strict limits.
- A *salesman* may take its own decisions and is not restricted in the way in which it intents to take care of its user's interest.

An agent may be able to perform all roles; a specific situation provides a context for an agent to perform a specific role, with associated autonomy in decision making. In the various chapters these roles are discussed when relevant and analysed given the legal rules and regulations that apply.

1.3 LAW AND LEGAL ASPECTS

To fully grasp the legal implications of software agents as discussed in the subsequent chapters, this section briefly introduces some basic characteristics of law, legal sources and relevant distinctions. In addition it will provide some differences and commonalities between civil law and common law systems.

1.3.1 Some Basics

The law can be found in several sources. Perhaps the best known source of law are statutes, that are brought about by national legislators. Every country has its own statutes. Statutes may thus diverge between countries. In order to diminish the divergence between the statutory laws of the individual national countries certain regional bodies aim to harmonize legislation. A key example here are the efforts of the European Union which has issued numerous so-called Directives. A directive is addressed to (the legislators of) the European Member states and obliges them to adapt a small part of their national legislation in such a way that it conforms to the directive. Unlike the word 'directive' might suggest, a directive is binding for the Member States of the European Union. Such adaptation of a national statute is called an implementation of a directive. The goal of a directive is as said mostly to harmonize national legislations (which is less far-reaching than a complete assimilation would be)

Law may also be found in international treaties or conventions. Such law is primarily addressed at the states that are parties of the treaty. Well-known examples are the treaties issued by the United Nations (UN) or the World Trade Organization (WTO). Usually these treaties are addressed to the government of a party-state. However, sometimes citizens can also directly appeal to an international treaty; a citizen may, e.g., invoke the right to respect his private life directly by appealing to art. 8 of the European Convention on Human Rights.

Statutes, directives and international treaties together can be referred to as 'written law' or 'statutory law'.

Apart from written law, another source of law are decisions by courts (often referred to as court rulings or case law). Sometimes the term 'law' is reserved for these rulings by the courts. The term 'law' may, however, also be used as a general term comprising all kinds of law. Caselaw together with so called customary law comprise so-called unwritten law.

Finally, something must be said as regards the terms 'illegal' and 'unlawful'. The distinction between illegal and unlawful is closely connected to the distinction between statutory law and case law. 'Illegal' means in contradiction with a statute, unlawful means in contradiction with unwritten law.

1.3.2 Civil law vs. Common Law

Two more general remarks to be made concern the difference between the civil law and the common law tradition, and the legal considerations regarding software agents.

In the way the law deals with software agents (and other) issues, the so-called common law and civil law traditions are distinguished. The common law tradition that is prevalent in the United Kingdom and the United States traditionally places much emphasis on case law (i.e., decisions by the courts), whereas the civil law tradition that is found in continental Europe attaches more value to statutory law. Over time, both traditions have grown towards each other: in common law countries the clear and easily accessible statutory rules have gained in relevance, whereas civil law countries have discovered that case law can provide some much needed flexibility. Nevertheless, the fact that both traditions have over a long time grown independently from each other has created two distinct bodies of law. For this reason, we will make hereinafter explicit for which tradition findings are valid, if the difference between the traditions is relevant.

The design and use of software agents raises many interesting legal questions and it is a good thing to be aware of the legal dimension when building or using a software agent. One must, however, not forget that software agents do not function in a vacuum. They operate in a technical environment in which other software and hardware performs its function and may equally cause and be subject to legal considerations/effects. A software agent may, e.g., rigorously defend the privacy of its user, while the underlying operating system sends personal information about the user to its builders. The designer of a software agent must take account of legal limitations, but he cannot and must not think that he is responsible for all legal consequences of its use.

1.4 A CONCEPTUAL MODEL FOR LAW AND COMPUTER SCIENCE

The ALIAS project is an interdisciplinary project, in which computer scientists and legal experts co-operate. One of the goals is to develop an understanding of the legal and technical issues at hand: a far from trivial task. This project illicits that for lawyers and technicians to work fruitfully together a common language or perhaps a common set of concepts is helpful, if not necessary. As this project focuses on legal preconditions for the design of software agents, a set of intermediary concepts has been defined that are both meaningful to legal experts and to computer scientists. The following concepts have been identified (Apistola, Brazier, Kubbe, Oskamp, Schellekens and Voulon, 2002a): autonomy, identity, traceability, integrity, and trust:

- *Autonomy* is the ability to act without direct interventions of agents (e.g., humans, software agents, etc.) or processes and to have control over ones own actions and internal state.
- *Identifiability* is the ability to know a name or other (usual) denominator of an entity, which is related to anonymity and pseudonymity.
- *Traceability* is the ability to recover the actions of processes or man.
- *Integrity* is an overarching concept that encompasses authenticity and originality. Authenticity is the ability to conform to the original, not stained by man or material made causes, a rather difficult feat in the digital world. Originality is the ability to distinguish between original and copy.
- *Trust* is also a generic term for a number of sub-concepts. Trust encompasses reliability, confidentiality, non-excessiveness and safety. Reliability of an agent process means that the process is able to meet its expectations logically uninterrupted for the total duration of its functional cycle. Confidentiality (also known as exclusiveness) denotes the ability to limit the availability of private or exclusive content to known authorized parties. Non-excessiveness means that the means are limited to just that those necessary to meet ones goals. Safety (also known as security) is an encompassing concept for the CIA interests (CIA is an abbreviation for Confidentiality, Integrity and Availability).

A number of parties are involved in the legal and technical analysis of the intermediary concepts. The parties involved with a software agent are the following: a designer (who creates agents), a producer (who manufactures agents), a supplier (who sells agents), an owner (who buys agents), and a user (who uses agents). Note that one human may have more than one role, e.g., one person may be both owner and user of an agent.

1.5 STRUCTURE OF THIS DOCUMENT

This document is structured on the basis of the intermediary concepts introduced in the previous section. Each chapter discusses an intermediary concept by analysing a number of recipes from both a legal and a technical perspective. Chapter 2 discusses an agent's autonomy, involving the ability of an agent to 'act' by closing a contract. Chapter 3 analyses an agent's identity and anonymity. Chapter 4 explores integrity and originality. Chapter 5 addresses the role of trust. Chapter 6 concludes this document with a discussion. The two appendices contain four of the scenarios used (Appendix A) and a description of the legal framework (Appendix B).

2 AUTONOMY

Autonomy in the context of this document refers to the characteristic of software agents that they function independently with little human intervention. Legally, autonomy raises interesting questions, such as: are the 'autonomous' actions of a software agent attributable to its user? If so, how can the possible negative consequences of such attribution be leniated? More technical are questions such as: is it possible to instruct agents in such a way that they autonomously do as bidded, but refrain from unwanted behaviour? What role does information provision play in these issues? Can agents be employed to close contracts on behalf of their users?

Two recipes in this chapter address the role of agents in contract negotiation. The third recipe addresses liability for faults in data-processing and the fourth addresses support of organisational processes. The first two recipes have appeared in a more condensed form as the paper "Are anonymous agents realistic?" in the LEA-2003 workshop (Brazier, Oskamp, Schellekens and Wijngaards, 2003a).

2.1 ILLUSTRATIVE SCENARIOS

For each of the recipes, illustrative scenarios are given. The first two recipes are about agents closing contracts on behalf of their user; one of the tasks of software agents on the Internet. The closing of a contract is subject to liabilities. In this, there is no distinction between the real world and the virtual world. Like in the real world, within the virtual world a distinction has to be made between the precontractual phase and the contractual phase. Simply stated, the pre-contractual phase concerns agreeing to make a contract, while the contractual phase concerns the actual closing of the contract. The latter two recipes are about an agent's functioning which is also subject to liabilities.

2.1.1 Pre-Contractual Phase

The recipe for pre-contractual phase is illustrated using the Chemical commodities market scenario Chemicality.com (see Appendix A). Chemicality.com is an electronic, on-line marketplace, in which agents represent involved parties. The marketplace offers a platform through which buyers and sellers can trade in basic chemical commodities such as caustic soda, solvents and acids. Chemicality.com uses a number of ways to facilitate trading in chemical commodities. First, all variables (such as grade, concentration, specs, delivery details) concerning the commodities are standardized. Secondly, both buyers and sellers are screened before they are allowed access to the marketplace. To ensure payment of the seller, credit insurance is offered. Thirdly, all communication is encrypted.

For the pre-contractual phase - the subject of this recipe - situation 4 of the scenario is of special interest. Negotiations not only occur between buyer agents and seller agents, but between co-operating buyer agents as well.

2.1.2 Closure of Contracts

The chemical commodities marketplace is again used to illustrate some of the concepts involved in contracting using agents. The product 'chemicals' is a complicated product; quantity, purity, other quality features, and its usability, require much communication between the seller and the buyer. Thus, the scenario illustrates the necessity and relevance of information duties between contracting parties. Furthermore, there is a lively market in chemical commodities; offers must be quickly accepted. It provides therefore an apt scenario to explain issues of offer and acceptance, which is legally the prevalent way in which contracts come about. The existence of 'third parties' that organise a market (e.g., an auction) of the commodities may serve to illustrate the complications of distributed responsibilities. Questions such as the following arise: if the organiser of the market provides the software agent with which parties are to trade, who is then to blame if the contracting is imperfect because of software agent failure? The producer of the software agent (perhaps causing the software bug), the organiser of the market (as the provider of the agent) or the user himself (not having checked its correct functioning)?

2.1.3 Liability for Faults in the (Data) Processing by Agents

Liability is always about accidents and things going wrong. In order for the reader to have an idea of what can go wrong when using agents a few examples drawn from the scenario's that accompany this report are described. What can go wrong?

In the hospital scenario, an agent manages a patient's dossier. One of its functions is to gather all data relating to the patient. If it fails to function correctly, the data about the patient may be incomplete; as a consequence the doctor may be ill-informed and may cause damage when treating the patient having trusted on incomplete data. Furthermore the data managed by this agent concern highly sensitive personal data. If the agent releases data about the patient to an unauthorised person, it may cause a breach of privacy. Finally the hospital scenario includes an agent that is responsible for knowledge management in a hospital. It has a vital function in making information (knowledge) available to the staff of the hospital.

In the scenario about the marketplace in chemical commodities, a software agent appears that negotiates on behalf of its user. If the agent is a bad negotiator (because of a fault in the programming or the use of an inadequate algorithm), the user may suffer damage.

2.1.4 Support of Organisational Processes

Both the hospital and local government scenario are suitable to illustrate the issues that arise in the context of this recipe. A hospital as well as a local government is a complicated organisation in which professionals work according to a strict division of tasks. The service delivery of either organisation requires a high degree of co-ordination between professionals. Software agents can be very helpful in scheduling the tasks of the professionals, because they can handle the complexity of allocating tasks to professionals, taking many constraints into account.

Furthermore, the hospital and local government are knowledge intensive organisations. The fields in which they are active are not static, but subject to constant evolution and development. Therefore, an expedient exchange of information about the disciplines in which they are active, about developments within the organisation and news about the sector is essential. Software agents can, as described above, play an important role in supporting knowledge management.

Issues of knowledge management and planning can both occur in open and in closed systems. Because these applications of software agents are still very 'cutting edge', this recipe focuses on closed systems (as is apparent from the choice of scenarios). Nevertheless, it is wise to remember that such applications can occur in open systems and that a designer may be more vulnerable from a legal perspective, because an open environment allows much less control of the context in which the software agent is to function.

2.2 RECIPE FOR THE PRE-CONTRACTUAL PHASE

Negotiations between parties are a means to agree upon issues that may result in the closure of a contract. However, not all negotiations lead to a contract; sometimes negotiations fail. In such a case, parties go their separate ways without further obligations. Sometimes, however, negotiations are not fully noncommittal. Costs have been incurred or expectations have been raised that, have, e.g., led to negotiations with other parties being broken off. In legal terms, this is the domain of precontractual liability; explicitly present in civil-law, but not in common law. This recipe explores means to channel negotiations to limit / prevent unpredictable implications.

Often negotiations between two (or more) parties precede a contract in which contractual obligations are made explicit. This recipe deals with the pre-contractual phase of negotiation; the next recipe describes the next phase, the contractual phase.

At the level of (abstract) legal rules the difference between the pre-contractual and the contractual phase is clear. Unfortunately, when it comes to assessing real-life cases the difference is often much more difficult to perceive. Consider, e.g., the fact that the form in which a contract is closed is generally not prescribed by law: contracts can be closed in many ways including orally, or even tacitly. That may leave the court (and lawyers) with the difficult-to-answer question whether talks about the subject matter of a contract must be qualified as pre-contractual negotiations or as the closure of a contract itself. When using software agents for the closure of contracts, similar assessment problems may occur; nevertheless, the difference in terms of legal consequences is so great, that it is unavoidable to deal with the distinction between the stages.

If negotiations are broken off during the pre-contractual phase a certain kind of liability can arise due to damages caused by this fact. These damages can, for example, consist of the costs involved in the preparation of an offer or in the lack of income incurred by not acquiring a (new) client.

When intelligent agents start behaving more and more autonomously, negotiations between agents, and negotiations between agents and humans, can become more and more complex. When an agent breaks off negotiations, the question rises whether this invokes liability: an agent may have fostered expectations that it or its user is obliged to fulfil. Another form of pre-contractual liability can occur when one of the negotiating parties fails to provide necessary information (known as the information requirement, see below) during the pre-contractual stage.

2.2.1 Legal Analysis about Pre-Contractual Liability

The legal analysis in this section is divided into three parts. First, the legal rules are explained, subsequently the parties involved and the measures they take are described and finally a conclusion is given under the heading 'legal requirements'.

2.2.1.1 Legal requirements during the pre-contractual phase

Two categories of requirements that are relevant in the pre-contractual phase are distinguished. First of all, negotiating parties may have to comply to a duty to disclose certain information. Secondly, the party that breaks off negotiations may be obliged to compensate damages the other party suffers as a consequence thereof.

Information requirements

In civil law systems, in general, there are duties to disclose information imposed on the negotiating parties because these parties have to behave in good faith. Each party is bound to disclose matters that are clearly of importance to the other party in making his decision, provided the latter is unable to procure the information himself and the non-disclosing party is aware of this fact. This disclosure should be of circumstances of which the other party is ignorant, when these can be determining factors to such party's consent, in the sense that had such party known of the existence of these factors, he would have offered different conditions or opted not to contract at all. As a more specific example, according to Dutch law, the following rules regarding information requirements can be distinguished:

- One has to take into account the justified interests of the other party;
- One has a duty to check certain information;
- One is generally allowed to rely on information given by the other party;
- Under circumstances, one has a duty to disclose certain information;
- A duty to disclose information (which rests on a seller) outweighs the duty to check certain information (which rests on a buyer).

Under common law, the starting point is that a party that has knowledge of relevant information is not obliged to disclose this information to the other party that does not have knowledge of this information. However, if a party *does* disclose information, then he must tell the truth, or risk being guilty of *misrepresentation*, which is a ground for the other party to break the contract. Non-disclosure of information, in other words: silence, cannot be considered misrepresentation. Misrepresentation can be *fraudulent* (knowingly or recklessly made with intent to deceive) or *material* (likely to induce reliance) or both.

Liability for damages sustained in the pre-contractual phase

There is no easy answer to the question as to when actual liability arises during the pre-contractual phase. However, the beginning of an answer might be found in the following. According to Dutch case law[1], there are three stages that can be distinguished within the negotiation process:

Stage 1: Parties are free to break off negotiations without any obligation to compensate the other party;
Stage 2: According to the criteria of good faith a party is still free to break off, however, under the condition that he compensates the other party for (all or some of) the expenses made;

[1] The law of each country deals with pre-contractual issues somewhat differently; in this section, the Dutch approach is described, that has developed over the years in case law. Although a Dutch court is less rigorously bound to precedents (i.e. earlier decisions of the courts) than is the case in the common law tradition, Dutch case law (and for that matter any civil case law) still is an important source of law. Especially, decisions of Supreme Courts (in the Netherlands: the Hoge Raad) are meant to bring unity in the decisions of the courts.

Stage 3: According to the criteria of good faith a party is no longer free to break off the negotiations. If that party nevertheless breaks off the negotiations, he is obliged to compensate the other party for expenses and - in some cases - for the profits the other party would have made.

As examples of the three stages the following can be mentioned:

Stage 1: The supplier of chemical commodities presents himself on the market as a supplier of chemicals. It is clear that he need not close a contract with each party that is interested in the commodities on offer.

Stage 2: A party is interested in specific chemicals in which there is very little turnover and that can consequently not easily be sold to another party. The party makes clear to the offeror that the delivery must be according to the 'just-in-time' principle. For the benefit of the interested party, the offeror orders the specific chemicals from its supplier. At a moment that the offeror can still cancel the order, the interested party withdraws. The withdrawal is allowed, but if the cancellation of the order involves costs, 'good' faith' may require the interested party to reimburse the costs of the offeror.

Stage 3: Comparable to the second stage, except that the interested party withdraws after the chemicals have been delivered to the offeror. The interested party cannot in good faith withdraw from the negotiations without compensation for the expenses, i.e., the costs of the chemicals.

In later case law, it was held that "negotiating parties are obliged to have their conduct determined by each others reasonable interests as well. Each of them is free to break off negotiations unless this would be unacceptable on the ground of the justified reliance of the other party on the conclusion of a contract or because of other circumstances of the case".

Depending on the specific factors mentioned above in civil law systems, a contracting party breaking off negotiations may be obliged to pay damages. These damages may consist of the expenses that are incurred in the preparation of the fulfilment of its obligation, or maybe even the loss of profits the other party would have made out of the envisaged contract.

Under common law, a party will not be liable for breaking off negotiations. However, as an exception to this basic principle, there are four grounds from which liability might arise.

1. An innocent party who relied on a "negligent misstatement" by the other party who led him to believe that a contract would be concluded, whereby the innocent party suffered loss and on the facts there was a special relationship between the parties. The offeror of chemical commodities assures, e.g., the buyer that he has more than enough chemicals of the desired quality in store. When it comes to contracting, he backs out.
2. It could be held that a "collateral contract" had come into existence. Suppose that in a certain case, negotiations are broken off, but one party has already started to provide services. A judge might hold that a contract was formed with regard to the service that has been initiated. Subsequently, a reasonable sum will have to be paid as compensation for the work performed. An offeror of chemical commodities could, e.g., already have begun to compose a synthesis of certain chemical substances for the benefit of a customer. If the deal does not materialise and the offeror is stuck with a composition that is without value on the market, the customer may have to pay a reasonable sum as compensation for the work done.
3. A party that incurs loss due to broken off negotiations might claim "restitution". This means that if a party carries out certain work which was not intended to be gratuitous but which was intended to be financed from the profit which the party would have made from the future contract. The offeror of chemical commodities calculates for the benefit of a customer what quantities are needed to make a certain composition. The cost of the calculation was intended to be financed from the profit/income generated by the sale of the chemical substances.
4. A party that, during the negotiations, in reasonableness may rely on certain promises of the other party, may enforce this promise. This is called "promissory estoppel". Please bear in mind that there are some major differences between promissory estoppel in Britain and in the United States. For more information, see: Cooke (2000), Fung (1999), Pham (1994) and Zwalve (2000). This may be the case if a party makes it appear as if a contract is certainly coming about.

2.2.1.2 Relevant parties and possible measures

When analysing the legal rules concerning pre-contractual liability, the first thing that becomes clear is the extreme vagueness of the rules. For instance, Dutch case law recognises three stages within the pre-contractual phase, but does not offer straightforward criteria to determine a possible liability in the second or third stage. From the courts' point of view, this is perfectly logical. Every case brought before them is different from another, therefore, vague, but flexible rules are preferable. In an attempt to make the general legal terms more applicable in a technical context, in this section the relevant parties and possible measures are identified.

The parties involved

The most relevant parties are listed. Other technical systems and players may be involved as well, such as builders of operating systems that have hidden applications, or middle agents (middle agents are mediators: agents that mediate between agents, e.g., see Dekker, Sycara & Williamson, 1997). It is, however, beyond the scope of this report to deal with these players too. It should be kept in mind though, that these parties can be held liable as well. The position of the middle agent is even more complex. Who can be held responsible for their actions? This subject needs to be addressed in the future. The parties that will be dealt with in this recipe are the following:

- the user of the software agent, i.e., the person on whose behalf the software agent is negotiating. The user of the software agent may be involved in two ways: 1. as the party that wants to back out of negotiations and 2. as the party that would like to close a definitive contract.
- the partner in negotiation, i.e., the other party with whom 'our' software agent is negotiating; since the position of this party is identical to that of the 'user of the software agent', it will not be dealt with hereinafter.
- the provider of the software agent, such as the vendor of the software agent or the offeror of the marketplace.
- the producer of the software agent.
- the provider of a marketplace.

The precautionary measures to be taken by the parties have been grouped by the parties.

The user of the software agent (that may want to back out)

- Warn the other party in negotiation at appropriate moments; e.g., if the other party in negotiation is about to incur significant costs for calculation, the first party may make clear that it may back out after the calculations are complete. If the first party informs the second party that the second party is the only party with whom the first party is negotiating, warn the second party that that does not necessarily mean that the first party is to close a contract.
- If the a party wishes the other party in negotiation to accept the first party's offer before a certain time, then this deadline needs to be communicated to the other party at the same time when the offer is communicated.
- If a party reaches a point at which the party decides to break off negotiations, that should be done unambiguously and immediately. The first party must communicate immediately to the other party that negotiations are finished without a contract.

The user of the software agent (the party that wants to close a definitive contract)

- Agree with the other party at appropriate moments about the status of negotiations: are both parties still in a preparatory stage or in the finalisation state? Example: if a party is about to break off parallel negotiations with other parties, make sure that the negotiations with an interesting party will lead up to a contract, by making that party explicitly express the status of the negotiations. If this fits in with a negotiation strategy (i.e., if it is a careful strategy), explain to the interesting party why this status information is needed to enlist co-operation.
- Strike a deal about a partial aspect, where appropriate. Example: if a party is about to incur costs (e.g., for calculation), make sure that the other party will reimburse the costs, even if the main contract does not materialise.
- Close a framework contract prior to negotiation, in which the issues surrounding the breaking off of negotiations are dealt with. In a framework agreement, the conditions under which negotiations may be broken off and the division of the costs between the parties can be more concretely dealt with; e.g., if the software agents are constructed in such a way that the transition to the finalisation phase is 'technically' clearly marked, a framework agreement can fix the legal significance of the 'technical' transition. This may mean that parties agree that between them, everything preceding the transition, legally belongs to the preparatory phase and that no legally binding 'commitments' occur prior to the transition. (See also hereinafter, under the 'provider of the software agent'.)

The provider of the software agent

- Warn the user that the software agent has negotiating capabilities that hold the risk that expectations may be raised that the user may not want to meet.
- If the software agent has technical measures that are meant to prevent that a conflict about finalisation of a contract arises, warn the user that such measures are not failsafe (especially if there is no framework contract regulating these issues; see hereinafter).
- If a software agent can be configured for more careful and more aggressive negotiation strategies, inform the user about the initial state in which the agent is configured.

The producer of the software agent

- Build the software agent in such a way that there is a clear distinction between the preparatory phase and the finalisation phase. The software agent and its negotiating partner must 'know' clearly what state the negotiations are in.
- The precautionary measures described for the first two parties above (i.e., the user of the software agent in his two roles) must be supported by the software agent; the software agent must warn its user at appropriate times so that the user can take sufficient measures or the software agent must take the measures itself (e.g., close a collateral contract about the costs of calculation).
- If a software agent can be configured for more careful and more aggressive strategies, inform the user about the initial state in which the agent is configured. Or construct the agent in such a way that it will only function after the user has configured a strategy.

The provider of a marketplace

- The provider must ensure favourable technical conditions for 'running' software agents.
- The provider can perhaps take the initiative to provide parties with a framework contract as described above.

2.2.1.3 Legal Requirements

As a concluding insight, the following can be remarked: what information has to be provided and at what time or stage this has to happen is dependent upon the circumstances of each case. This dependence on the circumstances of the case has profound impact on the design of agents and the protocols that control their behaviour. If an agent and/or its protocol is more tailored to a specific application, more contextual information is available that can be used to more precisely determine the information duties/needs during the consecutive stages. A buying-and selling agent is more general than an agent participating in auctions and such an agent may, in turn, be more general than an agent that is purely designed for one type of auction. The more specific the context is, the easier it is to determine the contents of the information duties and the times they have to be performed. Generally, more specific contextual information results in agents and protocols that are better able to adhere to the legal requirements its user may want to meet. On the other hand, a more general agent will function less perfectly and may make its user more vulnerable in legal perspective. Such vulnerability may be more undesirable if the reasons to comply with information requirements are more compelling. This may, e.g., be the case if the extent of possible damage is great (e.g., in case of an auction of valuable objects) or if the information duties serve to protect a weaker party (a court will apply the information duties more stringently).

As an upswing to the technical section, below a number of situations in which negotiations are broken off and that give rise to decommittment by a software agent, are described.

1. Consensual and explicit: both negotiating parties agree to decommit. This is the most 'peaceful' form of decommitment. Both parties conclude on their own that further negotiations are not viable and agree to stop the negotiation. E.g., the offeror of chemicals finds out that its chemicals cannot be delivered on time, and the buyer finds out that it cannot pay the offeror on time. Both parties now agree that further negotiations are not sensible.
2. Consensual and tacit: Both parties have agreed on decommitment conditions beforehand. If the agreed conditions materialise, decommitment procedures are automatically activated. Stock exchanges make use of this type of decommitment. When the market overheats due to the undesirable behaviour of automated buy and seller programs, rules activate to decommit all the participants in the market, effectively 'resetting' these programs.
3. Unilateral and explicit: One party communicates to the other party that it considers the negotiations to be finished. The background for such behaviour, may be that the negotiating agents have private deadlines, (Sandholm, 1999), based on which they decide to decommit. E.g., one of the parties decides that if an offer will not be accepted in two days it is in vain to pursue the negotiation. A possible explanation of such behaviour may be that the party in turn has a customer who has set an explicit deadline. Such a private deadline has, however, no value in law. If the other party accepts the offer after the private deadline of the offeror has passed, the offeror may still be bound by his offer. Whether the offer is still valid has to be decided on the basis of the circumstances of the case. For example, when trading biodegradable substances it is easily understood that it is not sensible to accept an offer to buy fresh materials at a later time than a few hours after the offer has been conveyed. It is reasonable to assume that such an offer is only valid for a few hours, even if parties did not agree this explicitly between them. The example shows again the value of a 'framework-' or perhaps better a meta-contract in which parties agree upon negotiation issues, such as breaking off negotiations. E.g., a down-payment has aspects of agreeing on meta-issues: when a customer shows the intention to buy a piece of furniture that is not in stock, the seller could ask for a down payment. This guarantees that if the customer decommits, the seller will not suffer too much damage. The down payment has paid for a substantial amount of the cost price associated with the piece of furniture.

4. Unilateral and tacit: agents have, e.g., stated deadlines in their negotiation. Decommitment then follows when a party decides to let a deadline expire. E.g., a seller assures one of its customers that the goods will not be sold until the next day. The potential buyer decides not to buy the goods. By letting the deadline pass the buyer shows its decommitment.
5. Merger: When a merger of the negotiating parties takes place, while negotiations are in progress. This situation seems rather exotic, but when companies expand it can be beneficial to merge with a company that provides services. In this situation it is of course no longer sensible to keep on negotiating, and thus it can be beneficial to extend the agent with the functionality to detect this situation.
6. Decommitment improperly so called: use of levelled commitment contracts (Sandholm, 1999). In this scheme the negotiation consists of several stages, each indicated with a separate sub-contract. Now when, e.g., the first subcontract has been fulfilled, either party can decide to stop the 'overall' negotiation.

2.2.2 Technical Considerations

From the legal analysis it has become clear that adequate communication between negotiators can prevent many problems. Letting the other party know what is important, prevents that conceptions about the subject and the proceeding of negotiations diverge. Technically, adequate communication translates into using well-defined protocols of negotiation. Information about costs of precontractual work, about parallel negotiations should be part of such protocols. Furthermore, which legal system holds and which general conditions apply and which obligations hold at which moment/during which phase, needs to be communicated/advertised in non-ambiguous terms. In addition, protocols may define specifics about message delivery (message is sent versus message is received semantics). The chemical commodities market may, e.g., require that all messages are guaranteed to be delivered to their recipients: reliability of communication is of paramount importance.

Languages such as the WSDL (Curbera, Duftler, Khalaf, Nagy, Mukhi and Weerawarana, 2002), OWL (successor of DAML-S and OIL; Patel-Schneider, Horrocks and van Harmelen, 2002), or RDF (W3c.org), but also most ACLs (e.g., Fipa ACL, 2002) upon which most communication between agents is based, currently do not include primitives for such qualifications, nor do they include properties describing the phases of negotiation/contract assessment. It is unclear if this is possible, since it would mean interpreting certain concepts and trying to put them in non-disputable formats, covering various jurisdictions. It is a topic of further research to discover whether this is possible: if not, to what extent this would require general directives concerning the use of software agents, including changes to legal systems. Open systems such as the chemical commodity marketplace may provide experimental contexts for this research, although closed systems (e.g., the grocery shopping scenario) may provide more initial structure.

There are logics in which, e.g., beliefs, intentions, commitments, good faith are expressed (e.g., Rao and Georgeff, 1993). Verification of an agent's status on the basis of such logics in which an agent's reasoning is expressed may be possible in a number of cases. The question is, however, whether courts of law are willing to consider an agent's intent/beliefs as trustworthy legal evidence. The implementation of such systems is, in most cases, not fully tractable nor verifiable, possibly making evidence unreliable.

It is not common for E-commerce protocols to distinguish between a pre-contractual phase and a contract closure phase; they usually focus on the latter phase (e.g., Sherif, 2000) in which non-repudiation and authenticity play an important role. Note that this may be due to the context of a specific legal system (or systems) in which these phases take place. Below, a number of protocols is briefly described:

- The Secure Electronic Transaction Protocol, (Lu and Smolka, 1999), focuses on secure payment card transactions (e.g., by Visa and MasterCard).
- The Internet Key Protocol (Bellare, Garay, Hauser, Herzberg, Krawczyk, Steiner, Tsudik, and Waidner, 1995) consists of a number of secure electronic payment protocols, with varying degrees of non-repudiation and authentication.
- The fair non-repudiation protocol (Zhou and Gollmann, 1996) is more general and involves a trusted-third party to ensure fairness about message reception and integrity.
- The open trading protocol (Bichler, Segev and Zhao, 1998) is a broader protocol for interoperability of electronic purchases, including payment, invoices and delivery.
- The Needham-Schroeder public key protocol is also more general and establishes mutual authentication between initiators and responders (Tanenbaum and Steen, 2002).

A view on the general and specific protocols described above is that their basic line may be used for assuring 'statements' between parties, and form a basis for protocols distinguishing the two phases. More research is clearly required on this topic, e.g., in the context of contract management systems, in which a pre-contractual phase is conceptually distinguished, but not operationally included (Boulmakoul and Sallé, 2002).

Time constraints in negotiation may, in some cases, be a point of concern. Technical guarantees for all real-time constraints are not currently feasible. If mechanisms for fault-tolerant agent systems, such as those being developed by (e.g., Marin, Sens, Briot and Guessoum, 2001), are included in new agent platforms a certain level of certainty may be provided for specific classes of constraints.

Agents and agent platforms (which support agents) may be obliged to trace their actions. Traceability involves logging information about agent actions, a process which leads to large amount of tracing data, possibly distributed among multiple agent platforms in different jurisdictions. For example, the chemical commodity market needs to trace important exchanges of messages, e.g., those dealing with starting and finishing a contract negotiation phase, in such a way that the information traced is both accurate and remains unchanged. Determining the granularity of the actions logged, the reliability of tracing data, storing and processing such tracing data, and the role of third parties may not be easily accomplished and requires more research.

Additional relevant technical viewpoints are discussed after the next recipe for contractual closure (Section 2.3.2).

2.3 RECIPE FOR THE CLOSURE OF CONTRACTS

The actual contracting by software agents raises other concerns: what is actually needed to agree on a contract? How can software agents technically comply with these requirements? Are there good practices to which a software agent can adhere to make the process of contracting run smoothly? The e-commerce market place offers astounding opportunities for the trade in goods and services. One can argue that entering such markets with automated help is almost a necessity to assist in the selection of offers and negotiate contractual obligations. Electronic agents can provide such assistance.

2.3.1 Legal Considerations

In the legal considerations of this recipe, two subjects are discussed. First, the question is addressed whether agents can validly conclude contracts. Secondly, the various kinds of information that is, or should be, exchanged during contract formation are described.

2.3.1.1 Legal background

Offer, acceptance and consent

In most legal systems, civil law systems as well as common law systems, a contract is formed by the 'offer and acceptance'-model. This model entails that when a party accepts an offer to enter into a contractual relationship, a contract is formed. Subsequently, the contract imposes obligations on the contracting parties. As an example the formation of a contract for the sale of chemicals can be mentioned. First, a certain quantity of a chemical is offered for sale, a price might be mentioned as well. Next, the other party may agree to the offer as it was made, thereby accepting the offer and closing the contract. Alternatively, both parties may engage into negotiations before a clear offer and acceptance can be distinguished. When the contract for the sale of goods is formed, two obligations arise: the one party is obliged to deliver the chemicals, the other party is obliged to pay a sum of money in return.

The exact properties that make up an offer or an acceptance differ per country. However, the United Nations Convention on contracts for the international sale of goods (CISG) is an example which, obviously, deals with contracts regarding the sale of goods and contains rules that occur in most legal systems. Article 14 subsection 1 of the convention states that:

> '[a] proposal for concluding a contract addressed to one or more specific persons constitutes an offer if it is sufficiently definite and indicates the intention of the offeror to be bound in case of acceptance. A proposal is sufficiently definite if it indicates the goods and expressly or implicitly fixes or makes provision for determining the quantity and the price.'

Furthermore, according to subsection 2 of the same article:

> '[a] proposal other than one addressed to one or more specific persons is to be considered merely as an invitation to make offers, unless the contrary is clearly indicated by the person making the proposal.'

This leads us to conclude that there are at least three properties an offer should bear in order to qualify as a legally valid offer:

- it should be *sufficiently definite*;
- it should indicate the *intention* of the offeror to be bound;
- and it should be *addressed* at one or more specific persons.

With regard to the definition of an 'acceptance', Article 18 subsection 1 CISG states that:

> '[a] statement made by or other conduct of the offeree indicating assent to an offer is an acceptance. Silence or inactivity does not in itself amount to acceptance.'

Both the offer and the acceptance require an *intention* of the offeror. Article 14 subsection 1 mentions the term intention, Article 18 subsection 1 mentions assent. In general, for declarations or promises to be binding, most legal systems require some form of *intention to be legally bound*. In general a software agent can be used for the expression of such intent (the consent). For this consent to be binding on the user of the agent, three requirements – specific for the closure of contracts with the help of a software agent - must be met:

- The user of the software agent has chosen to activate the software agent and to keep it active.
- The functioning of the agent was not subject to undue external influence, i.e., no hackers or fraudsters have influenced the functioning of the agent.
- The agent acted within the limits of its authorisations, which the user has set or which he knows about or should know about.

The acts of agents during contract formation are binding upon their users. Common law as well as civil law are equipped to deal with contract formation by agents (Weitzenböck, 2001).

2.3.1.2 Information exchange during contract formation

Introduction

Statements made during contract formation can have a (to be determined) legal effect, depending on their nature and the way in which they are communicated. These statements can be categorised as follows:

- offer and acceptance
- general terms and conditions
- additional information requirements

Offer and acceptance

As stated above, a contract is formed by offer and acceptance. This is the 'core information exchange'. Without offer and acceptance, there is no contract and no need to discuss other information exchanges. There has to be offer and acceptance first as a foundation for the other information requirements.

An offer needs to be sufficiently definite. The indication of assent to an offer amounts to acceptance. The users of electronic agents are bound by the actions of their agents.

General terms and conditions

'*General terms*' are clauses which have been drafted by a party to be included into a number of contracts (compare Article 6:231(a) Dutch Civil Code). The main question that needs to be answered is: how do general terms need to be presented for them to be a valid part of a contract? As legal systems vary, so do the approaches to answer this question. In general, three approaches can be distinguished (Cavanillas and Nadal, 1999).

1. The formal adhesion rule.
   According to this rule, the general terms must be included in either the offer or the acceptance, or there must be an express reference to them, or they must be available prior to contract conclusion;
2. The duty to deliver the general terms.
   Here, the general terms have to be delivered to the other party;
3. The red hand rule.
   According to this rule, surprising clauses will have to be presented clearly, they may not be hidden away in the small print of the general terms. The most viable solution is to explicitly include these clauses in the offer or the acceptance.

The European Union has chosen the following approach. According to Article 10(3) of directive 2000/31 on electronic commerce, contract terms and general terms and conditions provided to the recipient must be made available in a way that allows him to store and reproduce them.

Additional information requirements

Recent European directives place rather specific information requirements on contracting parties. This entails that one or both parties is obliged to disclose certain information before or after contract formation. The information requirements vary, depending on the question whether the other party is a consumer or not.

Information requirements and consumers

The European directive 97/7 on the protection of consumers in respect of distance contracts demands that consumers are supplied with certain information. The directive mentions two information requirements. First, 'in good time'[2] prior to the conclusion of a contract, the following information should be provided:

- the identity of the supplier and, in the case of contracts requiring payment in advance, his address;
- the main characteristics of the goods or services;
- the price of the goods or services including all taxes;
- delivery costs, where appropriate;
- the arrangements for payment, delivery or performance;
- the existence of a right of withdrawal (which is granted to the consumer by the directive but is subject to exemptions);
- the cost of using the means of distance communication, where it is calculated other than at the basic rate;
- the period for which the offer or the price remains valid;
- where appropriate, the minimum duration of the contract in the case of contracts for the supply of products or services to be performed permanently or recurrently.

After closure of the contract, the following information must be provided 'in good time during the performance and at the latest at the time of delivery of the goods':

- written information on the conditions and procedures for exercising the right of withdrawal;
- the geographical address of the place of business of the supplier to which the consumer may address any complaints;
- information on after-sales services and guarantees which exist;
- the conclusion for cancelling the contract, where it is of unspecified duration or a duration exceeding one year.

Some information that is to be provided under the first information requirement is allowed to be provided in the second stage, i.e., 'in good time during the performance and at the latest at the time of delivery of the goods'. The second information requirements demands that the information must be provided in writing or 'in another durable medium'. A 'durable medium' can be defined as 'any instrument enabling the consumer to store information addressed personally and specifically to him and which is mainly contained on floppy disks, CD-ROMs, and the hard drive of the consumer's computer on which electronic mail is stored.'

Information requirements and information society services

The European directive on electronic commerce is aimed at so-called information society services, this means 'any service, normally provided for remuneration, at a distance, by means of electronic equipment for the processing and storage of data, and at the individual request of a recipient of a service.' The e-commerce directive contains a number of information requirements. In this section, only those requirements that regard contract formation are mentioned. (In fact, one requirement has already been mentioned above regarding general terms and conditions). According to Article 10 of the directive, the following information must be given by the service provider (except when otherwise agreed by parties who are not consumers) clearly, comprehensibly and unambiguously and prior to the order being placed:

- the different technical steps to follow to conclude the contract;
- whether or not the concluded contract will be filed by the service provider and whether it will be accessible;
- the technical means for identifying and correcting input errors prior to the placing of the order;
- the languages offered for the conclusion of the contract.
- the service provider must indicate any relevant codes of conduct to which he subscribes and information on how those codes can be consulted electronically.

Other Information requirements

Apart from the information requirements that follow directly from directives that are specifically aimed at e-commerce there are also information requirements stemming from more general directives, including:

[2] When this 'good time' is, cannot be said in general. Parties must determine this dependent upon the specific context in which they find themselves.

- Terms must be plain and comprehensible (Directive on Unfair Contract Terms and the Directive on Consumer Sales and Guarantees)
- The selling price and the unit price must be indicated for all products referred to in art. 1 Directive on Indication of Prices. The prices must be indicated unambiguously, easily identifiable and clearly legible.
- The main characteristics of the goods and services must be provided under art. 4 of the Distance Selling Directive art. 4 of the Financial Services Directive.
- The consumer must be informed about the processing of his personal data (art. 10 and 11 Privacy Directive).

2.3.1.3 Considerations and recommendations for the designer of software agents

Who are the parties involved?

- The consumer.
- The supplier of goods and services.
- The provider of the software agent, such as the vendor of the software agent or the offeror of the marketplace.
- the producer of the software agent.

Note that a middle agent may also be involved, but that they are not further considered.

From the foregoing the following three types of requirements contracting parties will want to adhere to include:

- *Compulsory rules*: typically rules to protect recipients of services and consumers as found in Directive 2001/31/EC and Directive 97/7/EC. Such rules may also be found in Codes of Conduct (which are binding for members).
- *Basic conditions* for the validity of contracts: e.g., the condition that an offer is sufficiently definite.
- *Good practises*: "customary actions or code of behaviour" (Webster); e.g., storing a copy of a contract.

It will be the task of the producer of the agent and of the provider of the agent to ensure that the software agent supports the contracting parties in the legal requirements they have to meet and the good practises they will want to adhere to.

Compulsory rules

The compulsory rules as found in the directive on electronic commerce and the Directive 97/7/EC mainly concern information duties. From a technical perspective it is not so much the contents of these information requirements that is relevant; these are for lawyers to fill in once technical format has been found. It is rather the form in which and the time at which the information is to be supplied to the other contracting party that determine the technical decisions that are to be taken.

With respect to the form of information provision, the following requirements can be distilled from the directives:

1. Information must be provided in a clear and comprehensible manner in any way appropriate to the means of distance communication used (Art. 4 Directive 97/7/EC)
2. Written confirmation or confirmation in another durable medium available and accessible to him (Art. 5 Directive 97/7/EC)
3. Clearly, comprehensibly and unambiguously (Art. 10.1 Directive 2001/31/EC).
4. Contract terms and general conditions must be made available in a way that allows the recipient of the service to store and reproduce them. (Art. 10.3 Directive 2001/31/EC)

The second and fourth requirement require some additional explanation.

Ad 2.

The proposal for the Dutch implementation (Art. 7.1.9A.3 lid 2 BW) is:

> op duidelijke en begrijpelijke wijze schriftelijk of […] op een andere te zijner beschikking staande en voor hem toegankelijke duurzame gegevensdrager

This requirement does not exclude the use of electronic means [recital 13, Kamerstukken II 1999-2000, 26861, nr. 3 p.19-20]. However, not every electronic means will meet the requirements of durability, availability and accessibility.

If the consumer uses a mobile software agent for communication with the supplier, there may very well be a problem: is storage in a software agent durable? Is it available to the consumer if the agent is in somebody else's computer? Is the information accessible for the consumer in such circumstances? The answer to these questions is not clear, but an answer in the negative is not unlikely. The problem is aggravated by the fact that the directive burdens the supplier with the duty to provide the information in the required form, but he cannot order the

consumer not to use a mobile agent. Technically a solution may be found in that an ACL/agreed protocol provides for a way to indicate that information sent to the mobile agent of the recipient must immediately be forwarded to the user of the software agent and stored on a durable medium, such as the hard disk of the user. The supplier can then configure his information in such a way that the receiving software agent understands what is expected (forward to user and store on his harddisk).

Ad 4.

Certain information must be made available in a way that allows the recipient of a service to store and reproduce it. Again the question is whether storage in a software agent of the recipient legally qualifies as storage. Especially if the agent is mobile and resides on somebody else's computer system, the question of storage and reproducibility may from a legal perspective be answered in the negative. The answer can again be found in an expedient forwarding of the information and storing it on a durable medium.

The time at which information must be provided.

When going through the two directives the following cases can be found:

- in good time[3] prior to the conclusion of any distance contract (Art. 4 section 1 Directive 97/7/EC)
- in good time during the performance of the contract and at the latest at the time of delivery (art. 5 section 1 Directive 97/7/EC)
- prior to the order being placed by the recipient of the service (Art. 10 section 1 Directive 2001/31/EC)

To be precise these conditions concern the time at which information must be provided, in that they reflect the consecutive phases in the process of contracting. Technically, the process of contracting can be structured by distinguishing the following phases:

1. Negotiation: information is exchanged, parties are exploring whether an agreement is possible.
2. Finalisation: parties agree and close a binding contract
3. Execution: goods/services and money change hands.

The information duties then find a place in the structure (in the order they are described in the phases: '1', '2 or 3' and '1' respectively).

Basic conditions for the validity for contracts

The basic conditions for the validity of a contract concern issues that touch upon the contents of statements: the statement must be an expression of ones will, an offer must be sufficiently definite, a statement must reach the recipient in order to have effect, etc. It is therefore clear that communication plays a central role in contracting. Many legal problems occurring in the context of the closure of contracts find their basis in imperfections in the communication.

In order to minimise the chance of miscommunication, it is advisable to use agreed upon syntax and semantics for communication. In the field of international trade and EDI, some initiatives have been taken to come to some uniformity.

On the level of syntax:

- Electronic data interchange for administration, commerce and transport (EDIFACT): syntax approved and maintained by the UN Economic Commission for Europe (hereinafter: UN/ECE) for the facilitation of EDI. See: http://www.unece.org/trade/untdid/welcome.htm

On the level of semantics:

- Incoterms: terms devised by the International Chamber of Commerce (hereinafter: ICC) for the purchase and shipping of goods internationally. Incoterms predate the use of computers (they date from 1936), but have in 1990 and 2000 been updated to take the use of computers and networks into account (e.g., see http://www.glomato.com/UnitedShipping/glossary/incoterms.htm). The field of application of Incoterms is, however, limited to international shipping of goods.

We are not aware of the existence of semantics for contractual terms with a wider or more general scope.

Apart from these the three requirements specific for the use of agents in contracting must be met:

- The user of the software agent has chosen to activate the software agent and to keep it active.
- The functioning of the agent was not subject to undue external influence, i.e., no hackers or fraudsters have influenced the functioning of the agent.

[3] No general rules specifying when the 'good time' is can be given. Parties must determine the 'good time' dependent upon the circumstances of the case.

- The agent acted within the limits of its authorisations, which the user has set or which he knows about or should know about.

Good practises

Instruction of a software agent

- The user makes public for which contracts a software agent is authorised, e.g., through the use of certificates.

Generating evidence of the contract

The user configures his software agent in such a way that the following is ensured:

- adequate authentication of the identity of the parties and the contents of the contract.
- adequate logging of the final contract
- adequate logging of the preparatory documents; they often can be of help in the interpretation of the final contract.

What measures are to be taken by the parties discerned above with respect to conclusion of contracts?

The consumer:

- It is advisable that the consumer makes public (or at least communicates to partners in negotiation) whether the software agent is authorised for the specific contract. If it is desirable that the software agent autonomously closes more than one contract without human intervention, it is advisable to make known for what (type of) contracts the software agent is authorised.
- It is advisable that the consumer adequately checks the identity of the other party and the authenticity of the contents of the contract.
- It is advisable that the consumer stores the final contract
- It is advisable that the consumer logs events and declarations that occurred during the preparatory phase.

The supplier of goods:

- The supplier must make information available to the consumer at the right phase of contracting. For the various stages, see art. 4.1 Directive 97/7/EC (prior to conclusion) and art. 5.1 Directive 97/7/EC (during performance).
- It is advisable that the supplier uses agreed upon syntax and semantics for the contract.

The supplier of a software agent

- The supplier informs the consumer of the features the software agent possess.
- The software agent must be in a ‘safe’ initial configuration: all features that protect the consumer must be switched ‘on’.

The producer of a software agent:

- The software agent of the supplier supports structured negotiation that discerns the consecutive phases of negotiation.
- The software agent of the consumer must support immediate forwarding to the user, triggered by a message received from another (viz. the supplier). Casu quo, the software agent must be able to communicate to the supplier that it does not possess the forwarding feature.
- It is advisable that a software agent supports the use of syntax and semantics that is tailored to the needs of contracting.
- The user of the software agent must have control over the agent, in the sense that the user can determine when to switch the agent ‘on’ and ‘off’.
- The software agent must be protected against undue influence by third parties, such as hackers and fraudsters.
- It must be clear to the user of the software agent what the authorisations and the instructions of the software agent are.
- The software agent (of the consumer) must have means to communicate to another (the supplier) its authorisations and instructions.
- The software agent must be able to verify the identity of its user.
- The software agent must be able to verify the identity of the contracting partner and authenticity of the contract (e.g., through a digital signature).
- The software agent must be able to provide the contract and the loggings of the negotiations, so that they can be stored for safe keeping.

2.3.2 Agent Specific Technical Measures to Close Contracts.

This section describes the agent specific technical measures that can be taken to close contracts. The starting point is the actor-by-actor analysis of the legally important measures discussed in Section 2.3.1 of this recipe.

For most of the measures accepted techniques in, e.g., software engineering, human computer interaction and security, can be identified. This section only addresses agent specific measures. The Chemical marketplace scenario in matchmaking and automated negotiation situation are used to illustrate the technical measures.

2.3.2.1 Technical measures for the producer of the software agent:

The software agent of the supplier supports structured negotiation that discerns the consecutive phases of negotiation and can recognise and speak different protocols. In the chemical commodity marketplace, agents are equipped with the protocols required by the marketplace.

2.3.2.2 Technical measures for the producer of the software agent and the supplier of goods:

Software agents need to be able to structure negotiation and contract finalisation using a standard/agreed ontology of which the syntax and semantics is known.

Agents need an Agent Communication Language (ACL) to interact and provide agents the means to exchange information and knowledge. The origin of ACL's can be traced back to the Knowledge Sharing Effort (KSE) that was initiated by DARPA in 1990 (Labrou, Finin and Peng, 1999). In KSE researchers from both academia and industry co-operated. The goal of the KSE was to develop techniques, methodologies and software tools for knowledge sharing, reuse for design, implementation or execution. Nowadays a number of ACL's exist (e.g., FIPA ACL, 2002). For the development and use of agents it is important to know the characteristics of these languages. Standard ontologies are needed with which contract negotiation and finalisation can be structured.

In the chemical commodity marketplace, agents are equipped with the ontologies used to describe both the processes (closing contracts) as the products (chemicals). Ideally, one standard chemical ontology and one standard structured negotiation ontology are used in the marketplace, although different versions of these standard ontologies may be present at the same time. Agents may need to be (automatically) updated whenever a new version of a required ontology is published.

2.3.2.3 The producer of the software agent:

The software agent of the supplier supports structured negotiation that discerns the consecutive phases of negotiation.

There are three types of information exchange within the context of the closure of a contract: the exchange of information with the other party, the exchange of information with the marketplace, the exchange of information with parties that need to know information in order to fulfil their part in the collective that constitutes the contracting party.

In the chemical commodities marketplace, participating negotiating agents are required to be communicate with these parties, possibly involving specific protocols and ontologies, requiring translations between these protocols and ontologies.

2.3.2.4 The consumer

It is advisable that the consumer makes public (or at least communicates to partners in negotiation) for which contracts the software agent is authorised, and to what extent. In the chemical commodities marketplace, a participant may state that her agent is only allowed to buy caustic soda in relatively small batches, with a definite price maximum. More detailed information about her constraints may jeopardise her negotiation position (e.g., by stating price, quantities and budget limits).

2.3.2.5 The producer of the agent

It must be made clear to the user of the software agent what the initial authorisations and the instructions of the software agent are.

It is advisable that a software agent has the possibility to make its authorisations known to other parties (including other software agents).

The provider of agents for the chemical commodities marketplace needs to communicate the initial authorisations and instructions of its agents to (potential) users.

2.3.2.6 The agent platform

Support for authorisation should be provided by agent platforms. Different techniques may be used. Certificates is one option, encrypted messages using a PKI another (see Section 3.2.2.2).

Safeguarding of agents and their associated (contractual) data.

In the chemical commodities marketplace, agents which are hosted by the marketplace, need to safeguard the identities of their users, as well as their negotiation instructions, from both the agent platform and other agents. See Section 4.2 about Integrity and Section 5.2 about Confidentiality.

2.4 RECIPE CONCERNING LIABILITY FOR FAULTS IN THE (DATA) PROCESSING BY AGENTS

The autonomy of agents entails that they have a certain discretion to make decisions that involve the integrity of their data processing. An agent may, e.g., decide on which system or platform it chooses to fulfil its task. If its task is time critical (e.g., trading in stock) and the chosen platform or the network connections used, are very slow, accidents may happen that may lead to liability. For the parties involved in the use of software agents, the question is how to deal with the risk of liability because of faults in the data processing by the agent. It goes without saying that a high degree of autonomy does not make things easier.

2.4.1 Legal Analysis

A preliminary question, that is best settled first, is the question whether one may cause damage to another. In principle it is not forbidden to cause damage to another. A company will, e.g., undoubtedly prejudice its competitors, if it markets a quality product that all consumers want to have. After all, it attracts customers that without its intervention would have bought a product from a competitor. The competitors therefore suffer damage, but the company is not liable.

The causation of damage can only give rise to liability, if it conflicts with a rule of law, such as the imperative to act with due care, with due regard for the legitimate interests of others or to act not negligently. In practise it may very well be that most inflictions of damage to another conflict with such a rule and thus in practise, most inflictions of damage are capable of giving rise to liability. The basis of the reproach made to the inflictor of damage is often that although he should, he did not take precautionary measures that would have prevented the damage. The hospital scenario shows this: the person responsible for software agents that provide doctors with information about patients must take all the measures necessary to ensure that the right information becomes available to the doctor at the right moment. Because of the interest at stake (the physical well being of humans – there will be little room for error.

For some purposes – such as the protection of consumers -, the legislators have created stricter liability regimes. A stricter liability regime is created in that statutory law indicates certain factual circumstances: if damage occurs under the indicated circumstances a specified person is liable. It is not necessary to prove that the person acted negligently, without taking due care etc. It is enough to prove the causation of damage in the said factual circumstances. An example of such liability is product liability: the producer of a product is liable if the product by its defective design causes damage. The product liability regime is only applicable to damage that results in death or injury or in damage to another object that is usually meant for use in the private sphere and in fact has been used in the private sphere. It is not clear whether the product liability regime is also applicable to information products, such as software and software agents. If it does, designers of software agents must be extra alert when designing a software agent that could cause physical harm to persons or goods or information products. An example could be a software agent buying chemicals on the Internet.

With respect to damage that flows from unlawful data processing, Directive 95/46/EC gives a special rule (Article 23). The data subject who has suffered damage as a result of an unlawful processing operation or of any act incompatible with the national provisions adopted pursuant to Directive 95/46/EC is entitled to receive compensation from the controller for the damage suffered. He need not prove that the causation of the damage is attributable to the controller. But the controller may be exempted from this liability, in whole or in part, if he proves that he is not responsible for the event giving rise to the damage. This is not a strict liability regime; the burden of proof with respect to the responsibility of the controller has merely been reversed.

A software agent may not cause damage. To that end, precautionary measures have to be taken. The issue of precautionary measures raises two questions:

- Who has to take precautionary measures?
- To what length does one have to go when taking precautionary measures?

These two questions are dealt with hereinafter.

2.4.1.1 Who has to take precautionary measures?

In order to answer this question it is necessary to identify the persons that may have a role in the causation of the damage. The following persons can be considered:

- The designer or builder of the software agent. The program may contain a bug or the level of autonomy programmed into the software agent makes its behaviour unforeseeable.
- The user of the software agent: he configures the software agent and triggers its activities. He may be able to assess whether the agent can faultlessly interact with other agents or platforms.
- The party responsible for a platform on which the agent functions. The platform may inadequately accommodate the activities of the agent.
- The party making the agent available to the user, e.g., the vendor: he may have failed to warn the user about certain aspects of the software agent.
- A party relying on the functioning of the agent: he may light-heartedly have trusted that the functioning of the software agent was correct. Think, e.g., of a doctor believing that a software agent adequately manages a patient dossier.

The above may give the impression that it is easily detected who caused the damage. In practise, this is often far from easy, since all or most of the parties described above may be involved and to a lesser or greater extent have caused or contributed to the occurrence of damage. For example, an agent owned by A and produced by B, travels to a platform owned by C, where it is attacked by hacker D. The hacker introduces strange code in the agent, causing it to inflict damage to E. That D is a relevant causator goes without saying, but also the other actors may have contributed by not taking enough precautionary measures.

As a rule of thumb one can say that he who contributes to the causation of damage and can do something about it, must in principle do something about it. This brings us to the second question referred to above.

2.4.1.2 To what length does one have to go in taking precautionary measures?

In order to answer this question, first an actor by actor analysis of the precautionary measures that can be taken is given. There are many types of precautionary measures that can be taken, including:

The designer or programmer of the software agent:

- Verification that no programming errors were made.
- Provision of adequate documentation with the agent, that gives explanation about the functioning of the agent, indicates the purposes for which it may be used, warns for effects of agent usage that a user may not expect, etc.
- Verification that the agent is being delivered in a safe initial configuration.
- Providing access control measures.
- Providing a user interface that functions in an intuitive way
- Building self-checking features into the agent: the software may be programmed to check whether the platform it is used in is adequate for the function at hand
- Possibly providing self- (or remote) checking features into the agent: whether the agent functions correctly.
- Possibly providing integrity-checking into the agent, to detect tampering and other influences on its functioning.
- The possible applicability of the product liability regime means that a software agent designer must not think light heartedly about the measures described above.

The user of the agent:

- Verification that the user understands the way in which the agent functions and can be instructed.
- Verification that the agent is only used in a suitable environment (platform, other agents).
- Verification of the integrity of the agent.
- Prevention of unauthorised use of the software agent.
- Taking measures against collection of personal data from the software agent.

The system operator:

- Making sure that a (software agent) process is only allowed to run on a system if it fulfils pre-defined operational requirements.
- Verification of the integrity of the agent.
- Making sure that enough status information is available for the software agent process to continue on another system, when the software agent process is being terminated.

The party making the agent available to the user:

- Making sure that the agent is in a safe configuration at delivery time.
- Ensuring that all documentation is made available to the user
- Warning the user for unexpected effects of agent use.

The party relying on the functioning of the agent:

- Verification of the correct functioning of the agent, e.g., in case the software agent does something out of the ordinary, the user of the software agent could be asked to confirm the actions of the software agent.

Although at first sight, it seems quite plausible and straightforward to take precautionary measures, the decision about what measures are to be taken is not that simple. In the first place, precautionary measures come at a cost, if at all possible. The detection of bugs has to be performed by a programmer whose scarce time cannot be used for alternative purposes. For the production of documentation the same holds. Verifications while using agents slow down the pace with which the agent functions. In the second place, there is no natural limit to some precautionary measures. If a software agent has a high degree of autonomy it is hardly possible to foresee all situations the agent may find itself in. Research may enlarge the foresight, but it is hard to determine, if not impossible, if enough research has been done to have a complete overview of what could happen.

This leads us to the conclusion that the participants do not need to take every conceivable precautionary measure, but they merely need to take 'enough' measures. The new question therefore is: what does it mean to take enough precautionary measures?

This is the mirror image of the question: under what condition a participant is liable? Liability can be said to be the consequences law binds to taking insufficient precautionary measures. The question of the level of precautionary measures can therefore be sensibly studied by determining the conditions under which liability exists.

In law liability exists, if a person acts negligently (some others terms are in use as well). The word 'acts' here also comprises the omission of an act, such as the non-performance of a precautionary measure. What does negligence mean? Negligence is an open term.

The mainstay of non-contractual liability centres in most countries around an 'open' concept such as negligence, e.g.: in The Netherlands: 'onzorgvuldigheid'; in France: 'faute'; in the UK: tort of negligence. Somebody is liable, if he acts negligently and somebody else suffers damage as a consequence. The idea of such a central concept is stronger in some countries (France, The Netherlands) then in other countries (United Kingdom, Germany). In the latter countries, many 'wrongful' acts are specifically mentioned in a statute. The enumerative way of defining 'negligence', however, has its shortcomings; it soon appeared to be impossible to enumerate all wrongful acts. As a consequence, these countries to a greater or lesser extent accommodated an open concept in their laws. It is therefore reasonable to take an open concept as a starting point for our expose on non-contractual liability.

Although 'negligence' is a concept that is flexible enough to make it fit every situation, it is far from clear under what conditions somebody's act, can be said to be negligent. To a high degree, this is a problem that the laws in all countries have to deal with. In the way in which the countries deal with this problem, grosso modo, a common denominator can be found; four factors can be distinguished that are specifically meaningful, when it comes to determining negligence:

- The probability of the occurrence of damage
- The nature and expected extent of the damage
- The nature and benefit of the act
- The cost of taking precautionary measures

The application of these factors can be demonstrated with the help of the patient dossier in the hospital scenario.

What degree of care (precautionary measures) has to be taken with respect to the confidentiality, integrity and availability of the patient dossier and the data it contains?

A patient's dossier contains information that is used in the course of treatment of a patient. The doctor and other hospital staff rely to a high degree on the correctness, completeness and up-to-dated-ness of the data contained in the patient dossier. This information is one of the determinatives for the way in which the patient will be treated.

The probability of damage as a result of 'wrong' information is high, since medical treatments can have a profound impact on the health of the patient (to the better or to the worse). Furthermore, the damage concerns damage to a person (injury or death). This is a much more serious kind of damage than damage to property or financial damage. These two factors point towards taking a high degree of care when handling a patient's dossier or the information contained in it.

On the other hand, it can be argued that patient dossiers fulfil a beneficial role in the hospital. The quality of medical treatments is generally higher 'with patient dossiers' than 'without patient dossiers'. Furthermore, the role of agents in gathering information of a patient is a beneficial one. After all, hospitals often consist of many departments, with each department holding a piece of the entire dossier. With the automated help of software agents bringing together the entire dossier is much less labour intensive. So, where does that leave us with respect to liability? Liability should not be so stringent that patient dossiers and software agents managing the patient dossiers are discouraged at all. So this factor does not favour stringent measures. However, the benefit of a patient dossier is strongly correlated with the correctness, completeness and being up-to-date of a patient dossier. A high degree of precautionary measures is therefore indicated by this factor as well.

Finally, the last factor: the cost of precautionary measures. Precautionary measures are not a homogeneous commodity. Some differentiation is needed: a differentiation along the following line can be considered. Some precautionary measures can be programmed into the software agents; a software agent could, e.g., be programmed to regularly check for updates that a hospital department makes to its piece of the dossier. These measures have only a one-time cost, namely at programming time (network bandwidth and processing time are not considered). Their cost is therefore minimal and there can be hardly any reason why programming time measures should not be taken. Other precautionary measures are only possible through human intervention. For instance: one could go back to the patient and ask him to verify whether the information in the dossier is still correct. Since the cost of this measure is high, one can arguably say that measure need not be taken, unless there is a special reason to do so. For example, if the doctor suspects that the information is incorrect he must ask the patient or do a medical test in order to be sure of the correctness etc. of the data he uses.

The above example shows how the evaluation of the four factors can help in determining what measures are to be taken in order to avoid the occurrence of damage. The evaluation of the factors does (admittedly) not always provide an answer. What to do if, e.g., some factors point towards a high degree of care and others point towards a low level of care? The law does not always provide an easy rule about how to act in such a case.

There is however, one formula that can provide a guiding rule in such a situation: the learned hand formula. The formula stems from economics and seems therefore to be especially appropriate in case of property or financial damage; the grief that stems from personal damage cannot always be adequately accounted for in economics.

According to the learned hand formula precautionary measures have to be taken if their cost is less than the product of the expected frequency of the damage multiplied by the expected amount of the damage.

The value of this formula lies with the fact that it gives an indication of the interrelations between the (three of the four) factors. As a sideline, it must, however, be remarked that the practical usability of the formula is somewhat dampened by the fact the value of the variables involved (expected frequency etc.) may be hard to determine.

As a conclusion, the following can be said. The law of non-contractual liability centers around the key concept of negligence (this is the English term, the meaning may differ from country to country). 'Negligence' is the standard that is applied by lawyers. The four factors and the learned hand formula are aids for determining 'negligence' and they should be treated and dealt with as such. They can never replace the standard itself and their relative value should always be borne in mind when using them.

2.4.2 Technical Measures to Prevent Liability

This section describes agent specific technical measures that can be taken to diminish liability. The starting point is the actor-by-actor analysis of the legally important measures discussed in Section 2.4.1 of this recipe.

For most of the measures accepted techniques in, e.g., software engineering, human computer interaction and security, can be identified. This section only addresses agent specific measures. The scenarios in Section 2.1 are used to illustrate the use.

2.4.2.1 Measures to reduce liability for the designer or programmer of the software

Verification that the software provides the specified functionality.

- Adherence to standards (in development and implementation) may yield more reliable software, e.g., following ISO guidelines or adhering to specific paradigms and frameworks.
- Software verification is an area in which some progress is being booked. Verification of logical specifications is, in some cases, feasible. This, however, is not the same as verification of a running system. Verification that a safe initial configuration is indeed 'safe', is not likely to be possible either.
- Traces/audit trails may provide statistical evidence for particular behaviours of software, but do not guarantee correct functioning.

Verification that the agent is being delivered in a safe initial configuration.

The relevant parties are the agent platform owner (because its location is where an agent begins to compute); trusted third parties (because they are related to the delivery of the agent on or via an appropriate channel); and finally the programmer, (who is responsible for the correct procedures to start the agent). One agent specific technique that can be used here is the so-called 'agent-container' (Tripathi, Karnik, Vora, Ahmed and Singh, 1999; Noordende, Brazier. and Tanenbaum, 2002). This technique provides protection against tampering with an agent's state, and can be used to check if an agent starts in a specific, valid, state. This technique can be applied in all three scenarios (Section 2.1).

2.4.2.2 Measures to prevent liability for the user of the agent

Prevention of unauthorized use of the software agent.

Prevention of unauthorized use of an agent is currently a topic of research in the Agentscape project (Wijngaards, Overeinder, Steen and Brazier, 2002). A forthcoming report on available security techniques deals with this issue of unauthorized use of agent (IIDS, 2003) in which, e.g., the use of a Public Key Infrastructure (PKI) is analysed. Unauthorized use is a problem in all three scenario's (Section 2.1).

Chemical marketplace scenario

- An agent needs to be protected in such a way that tampering can be detected, and better: prevented.

2.4.2.3 Measures to prevent liability for the owner of an agent

Making sure that enough status information is available for the software agent process to continue on another system, when the software agent process is being terminated.

The agent container technique discussed previously (e.g., Tripathi, Karnik, Vora, Ahmed and Singh, 1999; Noordende, Brazier. and Tanenbaum, 2002) provides the vehicle to cater for the two measures to prevent liability in this section. The agent container can be subject to different requirements on migration (Picco, 2001). In heterogeneous environments, where different agent-platforms can host a mobile agent, the notion of weak migration must be applied to the agent container. Weak migration means that the execution state of an agent is discarded when it is migrated. Examples of agent platforms that support the notion of weak migration are Ajanta (Tripathi, Karnik, Vora, Ahmed and Singh, 1999) and Aglets (Lange, Oshima, Karjoth and Kosaka, 1996). In closed environments, with the assumption of uniform agent platforms, like the hospital scenario, the strong notion of migration can be used. Strong migration means that the execution state of an agent is added when it is moved to another agent platform. Examples of agent platforms that support the notion of strong migration are NOMADS (Suri, Bradshaw, Breedy, Groth, Hill and Jeffers, 2000), Ara (Peine and Stolpmann, 1997) and D'Agents (Gray, Cybenko, Kotz, Peterson, and Rus, 1997). This technique can be applied in all three scenarios (Section 2.1).

Protecting the informational privacy of the owner

Technical measures can be taken for ensuring that personal information only becomes available for those are entitled to receive it:

- limiting the personal information that a software agent contains to the minimum necessary
- encrypting personal information
- having a robust system of authentication for making sure that the software agent only (decrypts and) releases personal information to those entitled to receive it.

2.4.2.4 Measures to prevent liability for the person making the agent available to the user

Making sure that the agent is in a safe configuration at delivery time.

The agent container technique discussed above is applicable. Logging facilities also can be used to this purpose.

2.4.2.5 Measures to prevent liability for the agent platform supporting agents

An agent platform needs to explicitate what an agent may expect in terms of reliability, fault-tolerance and traceability. Provisions for fault-tolerance are usually made during the design of the software agent. An agent platform can only provide some support for reliability and fault-tolerance (e.g., by using agent containers to frequently store an agents state) and usually needs the co-operation of the software agent (e.g., to frequently provide its state to be stored). This is still a subject of research.

Traceability involves logging information about agent actions, a process which leads to large amount of tracing data, possibly distributed among multiple agent platforms. Determining the granularity of the actions logged, and storing and processing such tracing data may not be easily accomplished and requires more research.

Chemical marketplace scenario

- An agent may be provided with a 'contract' when it enters the marketplace, in which the conditions of use are made clear; only after explicit acceptance of the conditions of use the agent may participate in the marketplace.

2.5 RECIPE FOR THE SUPPORT OF ORGANISATIONAL PROCESSES

This recipe deals with two ways in which software agents can support organisational processes. In the first place, software agents can be used for scheduling the allocation of tasks to employees. Since the scheduler often has to take into account many constraints, the process of scheduling in general is rather complicated. The help of a software agent in 'finding' a (more optimal) schedule that meets the constraints may make life considerably easier. With the advent of PDAs and other electronic organisers, automated planning and scheduling of meetings becomes more commonplace, often involving not a central planner, but decentralised multi-agent planning processes.

In the second place, software agents can perform a useful role in knowledge management. Many organisations have come to the realisation that knowledge and its proper handling is vital for effective and efficient management of the organisation and have for that reason embraced 'knowledge management'. Since knowledge management is a relatively young concept, it is explained here in some detail.

Knowledge management caught on in the world after the publication of the 'The knowledge creating company' by Nonaka and Takeuchi (1995). Their founding work is the basis for this expose. Nonaka and Takeuchi differentiate between implicit and explicit knowledge. Implicit knowledge is knowledge as it is present in the 'head' of a person. Explicit knowledge is information stored on paper or any other medium. The main task of knowledge management is the creation of knowledge. This is done, by organising the relation and interaction between implicit and explicit knowledge. This relation and interaction can be characterised by four processes:

- Socialisation: the exchange of implicit knowledge,
- Externalisation: the conversion of implicit knowledge into explicit knowledge,
- Combination: the exchange of explicit knowledge, and
- Internalisation: the conversion of explicit knowledge into internal knowledge.

Software agents can play a vital role in facilitating knowledge management. Applied to the four processes the contribution of software agents to knowledge management can be described as follows:

- Socialisation: finding persons that are knowledgeable about a subject and providing a communication channel
- Externalisation: providing the means to create knowledge documents and store them in a database or store them in another way that makes them available for subsequent internalisation. Perhaps an (personal) agent could monitor somebody's activities and signal him when a moment occurs that is apt for externalising information. Thus, an incentive to provide information to the 'knowledge base' is given.
- Combination: finding external sources of explicit information that can be added or linked to the owner database and adding and linking this information.
- Internalisation: making information available in a timely fashion and in a way that meets the information demand.

2.5.1 Legal Analysis

Hereinafter, the legal implications of both tasks of software agents (planning and knowledge management) are elaborated on.

2.5.1.1 Labour law aspects of planning by software agents

In the hospital scenario, planning activities are crucial for the orderly progress of all processes within the hospital, but also for the position of the employees. Planning is, e.g., directly related to stress and work pressure, to function conformity of the activities employed, for autonomy in structuring an employees activities etc. It is therefore not surprising that from a legal perspective, the introduction of planning by software agents in an organisation is relevant. Generally, the introduction of automated planning is something that has to be done after consultation of the employees or their representatives, such as labour unions, representative advisory bodies, the works council. Dutch law provides the following rules:

- According to Dutch law, the employer (more precisely: the entrepreneur) has to provide the works council with an opportunity to advise him about the introduction of or change in important technological facilities (art. 25 section 1 sub k Wet op de Ondernemingsraden, hereinafter: WOR).
- According to Dutch law, the employer (more precisely the entrepreneur has to gain the approval of the works council for the following decisions: 1. regulations about working hours and holidays (art. 27 section 1 sub b.WOR), 2. regulations that have to do with employee assessment (art. 27 section 1 sub g. WOR), 3. regulations that have to do with employee discussion about the work agenda (art. 27 section 1 sub i. WOR), 4. regulations about the registration of, the dealing with and the protection of personal data about persons employed within the company (art. 27 section 1 sub k. WOR), and 5. regulations about facilities that are meant for or suitable for observance of or the supervision of the presence, behaviour or performance of persons employed within the company (art. 27 section 1 sub l. WOR).

The result of the advise or approval by a work council may seldom lead to an altogether ban on the use of software agents for planning purposes. A far more likely result will be that parties seek preconditions under which they believe the use of the software agents is useful on the one hand and justifiable with a view to employee interests on the other hand.

The preconditions for the use of software can be clustered in three groups. It concerns preconditions with respect to:

- the handling of (initial) input data, which will often be personal data as is argued later on,
- the planning itself, and
- the data generated during the process.

Here planning is not considered to be an action that takes place at a point in time before the employees go to work, but consider it rather to be a continuous process, constantly monitoring work progress and adapting the planning along the way. There exist many reasons for this: e.g., there could be too many random occurring events.

The three groups of preconditions are hereinafter used to describe the areas of attention with respect to planning agents. In its execution of planning the activities, the agent has to take possible agreements between employers and employees or their representative take into account. Even if in some jurisdiction planning agents are being used without prior consultation or agreement between the employer and employees, it may nonetheless be wise to bear the following points of attention in mind.

Handling of the input data

In order to fulfil its task a planning agent must have many data at its disposal about the employee whose work is to be planned. The following data can be mentioned: the function of the employee, his qualifications, his working hours/week, the days the employee takes a day off or has planned a holiday, whether the employee has reported ill and whether the employee has physical disabilities (if so, which?). These data will almost always be related to the individual employee and therefore constitute personal data.

Assuming that the use of these data for planning purposes has been given an adequate legal basis, the agent design and application still has to ensure that the data used are correct, complete, up-to-date and that they do not risk to fall in the hands of unauthorised parties. In this respect sufficient measures to ensure the use of the data have to be set in place. This aspect – however important – is something that is an instance of the more general problem of how to deal with personal data. This is a subject that does not lend itself for exhaustive covering in this chapter about autonomy; so for further information the reader is invited to read the chapters about integrity and trust.

The planning itself

The employee is generally obliged to perform the tasks the employer instructs him to do. This is the basis on which the employer is able to plan the activities that have to take place in his business or organisation and may instruct employees to execute the tasks that are ascribed to them in the planning.

The employer does, however, have responsibilities towards his employee. The employee should, e.g., not be exposed to unacceptable risks or dangers. This may be the case if the agent assigns the employee a task for which he is not qualified or if the employee is made to works too many hours without a break, and fatigue becomes a serious factor. The general rule that the employee should not be exposed to these risks and dangers is often made specific in regulations, such as collective labour agreements, internal regulations, safety regulations, etc.

Data generated during the process

If planning is a cybernetic process, a constant feedback about the progress of work and the availability of employees is needed. A planning agent will, e.g., keep track of the progress that employees make in fulfilling their tasks. For planning purposes, this information is needed in order to adapt the planning to changes in circumstances; e.g., an employee has finished his task in a shorter time than planned for or he needs more time. In either case the 'original' plan has to be adapted to the new situation.

From a legal perspective, it can, however, be remarked that the information about the progress of the work (that is gathered in the planning process) can be used for more purposes than just planning. Especially, data about individual work progress may be of interest to the employer on other grounds than the organisation of the workflow. The employer may, e.g., want to use the information for supervisory purposes: the information can in a later stadium be used to make decisions about employees (such as promotion, sacking, bonuses, reprimands etc.). These 'extra' uses of work progress information may easily invade the employee's informational privacy. At the same time, it cannot be denied that an employer can have a justified interest in knowing this information. Because of this (possible) conflict of interest, (representatives of) employers and employees have to come to an agreement about the circumstances in which an employer may have access to individual work progress information and about the form in which this information is to be supplied to him.

2.5.1.2 Legal aspects of knowledge management by software agents

As previously stated, knowledge management is not just about providing data (e.g., externalisation), but also about gathering data (e.g., internalisation). Nevertheless, the provision of data may give easier rise to legal problems than the gathering of data does, since it is directed at third parties or the public in general. Therefore, the gathering of data is not dealt with here explicitly.

This does not take away that in an indirect way the gathering of data is relevant: agents could be used to gain knowledge about the information needs of a user and could, based on this knowledge, proactively provide data to the user.

The legal problems that can occur in (pro-active) data provision may be twofold. On the one hand, there may be a problem with the data itself. The data may be incorrect, incomplete or untimely. Furthermore, the provision of the data could infringe upon an intellectual property right or the privacy of the person that is the subject of the data. On the other hand, there may be a problem in the determination of the information needs of the user.

Problems with the information itself.

If data is incorrect, incomplete or untimely, a person who trusts the data could incur damage. The same holds for provision of data that infringes intellectual property or privacy rights. The occurrence of damage is (of course) to be prevented; so the question arises as to who has to take which precautions to avoid damage. This question has been dealt with extensively in the previous recipe; therefore, this question need not be answered again here.

Problems with the determination of the information needs of the user.

Information retrieval is an acknowledged difficult terrain in Computer Science. Many problems are still unsolved (see hereinafter), resulting in systems that cannot be guaranteed to function completely adequate, in the sense that information needed will also be found. Therefore, it is paramount that builders and providers of information retrieval systems explain clearly what the limitations of their information retrieval systems are.

2.5.1.3 Legal Requirements

Planning

Who are the parties involved?

- The employer/ the user of the software agent.
- The employees who's activities are being planned by the agent.
- the provider of the software agent, such as the vendor of the software agent or the offeror of the marketplace.
- the producer of the software agent

What kind of requirements can be distilled from the foregoing legal expose? Upon examination of the foregoing, it appears that the legal considerata can be subsumed under three headings, visually: accountability, self-regulation and vulnerability.

Accountability

The interests of the parties involved with respect to the planning process are great and thus the actions of the software agent must be accountable. For the purpose of accountability, it is important that the functioning of the software agent is transparent.

- On the one hand, this concerns ex ante transparency. It must be possible to know beforehand, what (kind of) activities the software agent will be undertaking.
- On the other hand, this concerns retrospective transparency. Afterwards, it must be possible to retrieve what the software agent has done. There must be logging trails.

Self-regulation

The parties discerned (the employer and the employees) have each their own set of interests involved in the planning process. The interests may in certain respects be incompatible. The way in which such incompatibility is resolved is not always fixed in statutory law or caselaw: it is often left to the parties concerned to find modalities for planning that are acceptable to both sides. This means that the solutions that are to be found can differ from organisation to organisation.

- For a software agent, this means that it must be flexible; at least the initial configuration must be adaptable to the specific context of the organisation in which it is to function.
- The solution that is found between the interested parties may not have unlimited validity. Results may be renegotiated and the agent must be able to adapt its activities in order to accommodate the newly found modalities. The agent must thus be reconfigurable.

Vulnerability

The process of planning involves critical information, such as personal information. The information used by and generated by the agent may be interesting for third parties.

- For the agent design this means that sufficient measures must be taken to prevent unauthorised access to and use of the information involved in the planning process.

Knowledge management

Who are the parties involved?

- The provider of information
- The consumer of information
- The producer of the agent that works either for the provider or the consumer of information
- The management of the organisation in which knowledge management applies.

With respect to the information to be used several issues arise:

- issues with respect to the correctness, completeness and up-to-datedness of information and information systems.
- Issues with respect to the status of the information: business secrets, intellectual property rights and privacy.

The correctness, completeness and up-to-datedness of information

The explicit knowledge in databases, will often be entered by human beings. For a software agent, it is complicated to check the correctness, completeness and timeliness of information entered. Such would require a level of understanding of natural language and a level of 'background knowledge' against which to check the newly entered information, that is not realisable or in many cases economically unfeasible.

- Software agents do have a function in guarding the availability, integrity and casu quo exclusivity of information, once it is available in digital form.

- As a good practise, a software agent could provide an invitation to humans entering new information to verify its correctness, completeness and up-to-datedness.
- As a good practise, software agents can point out to humans that enter information into a system that they also provide meta-information about 'how to use' the information, or how the information came about, so that the potential user of the information can better judge the value of the information or the purposes for which it can be used.
- As a good practise, software agents can point out to humans that enter information into a system that they take notice of the privacy policy before entering new information in the system.
- If a software agent has a function in information retrieval, the following is relevant. Information retrieval is difficult and imperfect. This may imply that it is more than in other respects necessary to try and conform as much as is possible to the state-of-the-art and to warn the user for known imperfections.

The status of information

- The software agent must guard the exclusivity of information that touches upon privacy or is a business secret.
- If information is covered by intellectual property rights, the license management may require that the information is only made available to a restricted group of persons. A software agent may have a function in guarding the information by requiring users to identify themselves or show their authority-to-use.

2.5.2 Technical Considerations

This section discusses technical considerations for planning and knowledge management.

2.5.2.1 Planning

For a software agent, this means that it must be flexible; at least the initial configuration must be adaptable to the specific context of the organisation in which it is to function.

The solution that is found between the interested parties may not have unlimited validity. Results may be renegotiated and the agent must be able to adapt its activities in order to accommodate the newly found modalities. The agent's adaptation may require re-configuration of the agent.

From a technical perspective, it can be remarked that a planning agent does not necessarily provide the possibility to report about (the work progress of) individual employees or provide data from which easily conclusions about individual employees can be drawn. The information may only be useable for planning purposes, or the possibility of a management report about work progress is only available in an aggregated form. The form in which information is available (on the individual level, subject to policies that compartmentalise data, aggregated or not at all) is the result of a design choice. What design choice is made has a profound legal significance.

If the software agent is constructed in such a way that information about individual employees cannot be made available, the employer is placed before a 'fait accompli'. On the other hand, if the information on the individual level is available, the temptation to use it will be close to irresistible. Argumentation that comes down to 'not using information that is available' often has a hard time; phrases such as the following can be heard: 'Do employees have something to hide?' and 'If you do your work well, you have nothing to fear'.

Technically planning agents should thus be able to communicate with so called data protection agents (Serban, 2002a). Their planning will then probably not be perfect but will not disagree with data protection issues.

If planning agents are built with standard agent software, it is perhaps wise to build a compositional designed software agent (Brazier, Jonker, Treur and Wijngaards, 2001). As a pragmatic solution, the reporting facility could, e.g., be programmed in a separate module that is only delivered upon special request in specifically determined circumstances.

The agent platform(s) involved need to maintain logs (traceability) and provide sufficient continuity and reliability, see Chapter 5.

2.5.2.2 Knowledge management

Information retrieval is difficult and imperfect. This may imply that it is more than in other respects necessary to try and conform as much as is possible to the state-of-the-art and to warn the user for known imperfections.

As mentioned above, information retrieval suffers from a number of shortcomings. In literature (Matthijssen, 1999) the following technical methods, with their shortcomings, are described:

- Indexing: the index the information contents of documents only partially.
- Query formulation: A query is an imperfect description of an information need
- Matching: The matching function operates rough heuristics and a tight closed world assumption
- Conceptual gap: the discrepancy between the users view of the subject matter of the stored documents and the reduced formal view on these subjects as presented by information retrieval systems.

In the local government scenario, agents may need to dynamically reconfigure their way of working when, e.g., an increase occurs in the demand for specific information: agents of other employees may be informed of such information, and its use, before even their human counterpart has formulated requests. When information changes, agents need to quickly and efficiently propagate these changes to those agents and humans currently working with the information, as well as ascertaining which previous records/transactions/etc. need to be reviewed in the light of the changes to the information.

The software agent must guard the exclusivity of information that touches upon privacy or is a business secret. Technically agents can be used to solve this issue (Serban, 2002). Using agents to guard the use of data allows explicit policies that can be negotiated by the organisation and its employees. Furthermore this means that agent communication languages (ACL's) have to be developed that allow for exchange of data protection related information about data. The Fireball Modelling Language (Serban, 2002) defines a meta-language to model the activity of such agents, and gives designers a framework that helps to build such an agent communication language.

2.6 DISCUSSION

Contracting plays a role in many possible applications of software agents. E-commerce is an obvious example, but by far not the only one. This paper focuses mainly on applications within e-commerce, but many of the observations also hold for other contract forms.

What has become obvious is that closing a contract with the help, or even by means of a software agent raises a number of questions from a legal point of view. The legal rules are all directed towards human beings or corporations. Using software agents means introducing a new and different legal situation. Not all legal rules can be applied to that situation. But also software agents have to be constructed in such a way that they meet the legal requirements for closing of contracts.

This chapter has named some of the questions about the legal requirements for legally valid closure of contracts by software agents. It has also addressed some of the technical possibilities to meet with these requirements. It is important to note that this paper only addresses and poses the questions and does not yet give full solutions to the problems indicated. For this a lot more research is necessary as often stated in this paper. It is, however, clear is that software agents also need to distinguish between the precontractual and contractual phase, since the obligations, liabilities and according requirements, both technical and organisational, differ in these phases. It is unclear whether a software agents will ever be able to distinguish between these phases on its own. The border between these phases is rather diffuse and dependent on the interpretation of concepts like good faith. The problem may be partly solved by agreements and standards incorporated in generally agreed upon protocols for agents. It remains to be seen whether such solutions can also be used to handle the difference between legal systems. We would strongly recommend to research the possibilities to reach agreements on these topics and incorporate those results in negotiation protocols.

Other topics that need to be researched more closely are the necessity for traceability of an agent's actions referring to evidence in negotiations. How secure and durable should this information be? To keep all information may lead to an information overload. What is legally necessary, minimum and maximum? Can standards for this be defined? Can this information be secured, so it cannot be interfered with in a later phase, thus validating its authenticity to make it valid legal proof?

These and many more questions will actually rise when agents roam the Internet and are able to close contracts. Now it is the time to research and explore these topics to formulate the measures needed in the future.

3. IDENTIFIABILITY AND TRACEABILITY

The ability to identify and trace seems to be almost indispensable in society. On the one hand, people need to identify and trace other persons, whereas on the other hand, the persons concerned may have a justified interest in escaping from these identification and tracing attempts. Software agents can have a function in both identification and tracing (e.g., shopbots) and in hiding identity and the frustration of tracing attempts (e.g., anonymous remailers). The two first recipes in this chapter deal with identity and traceability in the relation between the person concerned and the person (trying to) uncover the formers identity. These two recipes appeared in a shorter form as the paper "Are Anonymous Agents Realistic?" at the LEA-2003 workshop (Brazier, Oskamp, Prins, Schellekens and Wijngaards, 2003). The third recipe deals with third parties that provide services for the hiding of ones true identity.

3.1 ILLUSTRATING SCENARIO

The two scenarios involving trading chemical commodities and shopping for groceries are used to illustrate identifiability and traceability: each recipe is illustrated by an example taken from the chemical marketplace scenario or the grocery shopping scenario.

3.1.1 Identification

A software agent approaches a chemical commodities marketplace somewhere on the Internet and closes a contract for buying caustic soda. After payment has been completed, the software agent and its human counterpart wait in vain for the caustic soda to be delivered. After some deliberation, the human counterpart decides to contact the holder of the marketplace. However, the software agent has not stored at which marketplace it bought the caustic soda. The user of the software agent wonders whether the marketplace had made its identity known to the software agent. If it did, in what form had the marketplace revealed its identity? Should it not have been revealed in such a way that the buying party had the identity at its disposal at the moment it needs it most? Are there protocols for such situations? And should perhaps the producer of ‘our’ software agent have taken responsibility and provided a way to store identities?

3.1.2 Anonymity

The shopping cart does not register the identity of the customer; the customer can use a coin to operate the cart (instead of a chipcard). However, the grocery store uses video-cameras to register customer behaviour. Customers may be linked to the goods bought and the payments made, through combining data in the cart (goods bought) and video-images (showing the customer and the goods that are in the cart). Anonymity inside the store is not easily achieved: although no identities are used, customers may very well be traceable and perhaps even identifiable. The store may not allow completely anonymous customers (e.g., with regard to shoplifting). The human customer is, in this situation, an identifier for itself: although he or she may change his or her appearance, store personnel and/or video equipment may be able to recognise a returning customer. If at some point in time the customer reveals his or her identity, the store can retroactively analyse this customer's behaviour… Is anonymity only temporary?

3.1.3 Facilitation of Anonymity

The holder of a chemical commodities market place allows deals to be made pseudonymously as economically strong buyers and economically weak sellers desire this feature. ‘Our’ software agent buys chemicals at the marketplace from an anonymous seller. After the deal is struck identities are exchanged to arrange for delivery. However, the identity provided by the seller appears to be incorrect. The chemicals are not delivered. Our software agent decides to ask the holder of the marketplace to give the identity, but the only identity he can provide is the false identity our software agent knows already. Should the holder of the marketplace have more thoroughly checked the identity of the seller? Or is it the responsibility of our software agent not to do business with an anonymous seller to avoid nasty surprises?

3.2 RECIPE FOR SELF-IDENTIFICATION

Both legally and technically, it is beneficial to have identities. Since the law primarily regulates relations between humans, identification or identification duties legally often concern humans. The identity of inanimate objects is however not completely void of legal interest. It may, e.g., be of interest for finding the facts about relations between people.

From a Computer Science perspective the emphasis is much more on the identity of the software agents themselves. An agent platform uses process identities to distinguish between different agent processes. Software agents may have separate identities for communicating with each other. An identity of a user is however also of interest: if, e.g., access is to be given to data stored in a software agent it is relevant to identify the person authorised to do so (although strictly speaking, determination of an authority need not encompass determination of an identity).

What identifiers qualify as an identity depends on the context, especially the purpose for which the identity is to be used. Legally, a name, address and dwelling place amount to an identity, e.g., because this allows to subpoena somebody. Technically, any unique identifier may suffice: it may be more important that an agent platform can dynamically allocate an identity than that the identity consists of pronounceable name.

3.2.1 Legal Analysis

The meaning of the term identity is related to the term 'anonymity', which is characterised by the fact that other parties do not know one's identity. Froomkin (Froomkin, 1995; 1996) distinguishes between four types of electronic communication in which the sender's physical (or 'real') identity is at least partly hidden:

1. Traceable anonymity: the recipient does not have any clues as to the identity of the sender. This information is in the hands of a single intermediary;
2. Untraceable anonymity: the sender of an electronic message is not identifiable at all;
3. Untraceable pseudonymity: this is similar to untraceable anonymity, however, here the sender uses a pseudonym. A pseudonym differs from an anonymous denotation in that pseudonyms can be used to build up an image and a reputation just like any other online personality. Therefore, pseudonyms are used consistently over a certain period of time, while an anonymous denotation might be used only once, for a single message;
4. traceable pseudonymity: this involves communication using a pseudonym which can be traced back to the sender, although not necessarily by the recipient. Within the category of traceable pseudonyms, one could further distinguish between pseudonyms that have been accorded 'formally' to somebody by a 'third party' or pseudonyms that have been chosen by the holder of the pseudonym him- or herself.

Apart from these four types of communication where the sender's identity is hidden in some way, there is the situation in which the sender is using his/her 'real' identity: there is no anonymity or pseudonymity.

Identifying oneself is compulsory if a statutory identification duty exists to do so. Often however, identification takes place in societal relations in which a duty to identify is not at stake. In this legal analysis, statutory identification duties are dealt with first. Subsequently attention is paid to the generally non-compulsory identification in the context of closing contracts.

3.2.1.1 Identification duties

A distinction of identification duties.

Identification duties can be divided into active and passive identification duties. In case of a passive identification duty, one only has to identify oneself after being asked to do so. In case of an active identification duty, one has to make the identification information available, whenever one is in a situation in which one is subject to the active identification duty. For example, whenever one takes part in traffic with a car, it has to carry a licence plate.

Is there a general duty to identify oneself?

In the Netherlands, there exists not a general duty to identify oneself [Tweede Kamer 2001/02, 28069, nr. 1, p. 6-7]. It is, however, a criminal offence to indicate a *false* name, Christian name, date of birth, 'official' or factual place of residence, when asked for ones identification data by a competent authority (Art. 435 Dutch Criminal Code). The Dutch government is, however, working on a proposal for introducing a general identification duty in the Netherlands. (http://www.justitie.nl/pers/persberichten/archief/archief_2002/131202concept_-wetsvoorstel_voor_een_algemene_identificatieplicht_klaar.asp) Plans to do so have been circulating for some

time (NOS, 2002). In a number of other countries, a general legal identification duty already exists (Steen, 1988).

Are there duties to identify oneself in special circumstances?

The non-existence of a general duty to identify oneself does not mean that there cannot be a duty to identify oneself in special circumstances. Hereinafter, a number of relevant special circumstances are dealt with.

Special circumstance: when using the Internet?

The Dutch police proposed to give every Internet user a 'licence plate'. This proposal has met with much criticism and will as a consequence no longer be pursued by the police [De Telegraaf, 1999; De Gelderlander, 1999].

Special circumstance: when using a software agent?

Is there a duty to make a software agent identifiable? Just like cars have to have license plates, it is conceivable that also software agents should carry some 'tag' that enables their identification in certain circumstances. The reason for such a duty could arguably be that software agents – because of their capability to act – can cause damage, while the responsible user may not be traceable; traceability is after all complicated because of the mobility of the software agent. It may very well aid a victim who is seeking redress for harm if the software agent carries a tag identifying its user. However, such a 'duty to make identifiable' (that could perhaps rest on the user of the software agent) does – to our best knowledge - not exist.

Special circumstance: applicability of the directive on electronic commerce (2000/31/EC).

The directive is applicable to providers of the so-called services of the information society. These are services normally provided for remuneration, at a distance, by means of electronic equipment for the processing (including digital compression) and storage of data, and at the individual request of a recipient of a service. The holder of an electronic chemical marketplace is, e.g., a provider of services of the information society. In its fifth article the directive obliges service providers to make identifying and contact information available to the recipients of their services. This information includes inter alia: the name of the service provider, his geographic address, his e-mail address and his trade register number. About the form in which the information must be supplied the Directive stipulates that the information must be 'easily, directly and permanently accessible'. In the preparatory documents to the proposed Dutch Implementation Act, it is stated that the requirement of permanency can be met by placing the information on a website that is accessible for the general public. A provision of information at a point in time (such as is the case with e-mail) is not sufficient. The requirement of an easy and direct access can be met by presenting the information and the indication of its location in a way that is clear and recognisable to the users. The requirements also hold for mobile communication. If the small display screens of mobile devices do not allow to present the information in the required way, the Dutch government advises that the service provider refers to a website where the information can be found [Tweede Kamer 2001/02, 28197, nr. 3, p. 37-38]. With respect to commercial communications (a.k.a. spam) the directive stipulates that the sender (more precisely: the person on whose behalf the commercial communication is made) has to be clearly identifiable (art. 6 Directive 2000/31/EC). According to the proposal for a Dutch Implementation Act, this requirement can be met by providing a reference to the Internet-address where the identificatory information can be found [Tweede Kamer 2001/02, 28197, nr. 3, p. 42]. It is clear that agents acting as service providers or agents making commercial communications have to comply with these requirements. Considering that an agent is not a persona in law, the agent has to make the identity of the user available.

Special Circumstance: applicability of the 'Wet op de identificatieplicht' (Dutch Identification Duty Act)?

The Dutch Identification Duty Act introduces a passive duty to identify oneself in certain circumstances. These circumstances concern: labour relations, social security, financial services, request for a social security number, registration with an employment office, at the notary, as a fare dodger and visiting a soccer match and in the context of supervision over aliens. The identification duty is formulated as a duty to hand over, on request and at once, ones identity card to whoever asks for it in the legitimate exercise of his task [Tweede Kamer, 2001/02, 28069, nr. 1, p. 2 (Notitie beperkte uitbreiding identificatieplichten)]. The Act (or an implementation decree) indicates exhaustively which means of identification (ID cards) can be used to fulfil ones identification duty. These means of identification are: official travel documents, such as passports and documents which an alien must have at his disposal according to the Aliens Act to enable the determination of his identity, nationality and residency status (art. 1 Wet op de identificatieplicht). If ones nationality is of no concern, also a driver's licence can be used to prove ones identity (Hofman-Ruigrok, 1994: pp. 1538). Since none of these documents lend themselves to be 'handed over' in an online situation, it seems that the 'Wet op de identificatieplicht' is only of theoretical importance for on line situations. Even if an identification document contains a chip that holds ones

name and other identificatory information, identification at a distance is only possible if there is a reliable infrastructure for the reading and transmitting of these data to the verifier that sits at a distance. As long as such chips or infrastructure are lacking, a duty for identification at a distance is pointless.

A legislative proposal is pending that allows biometric features to be incorporated in travel documents [Tweede Kamer 2001/02, 28 342 (R 1719), nrs. 1-2, Wijziging van de Paspoortwet, onder andere in verband met het toepassen van biometrie in reisdocumenten]. The presence of a biometrical feature makes it possible to reliably check whether the holder of a travel document is its rightful holder. This also opens up the possibility to use a travel document for reliable identification at a distance, e.g., on the Internet. The proposed statute is, however, still primarily concerned with enrolment and the issuing of travel documents with biometrical features. Questions regarding verification and a possible duty of the holder to co-operate to a biometrical verification are not addressed by the proposed Statute. At the same time, identification duties are still formulated in a way that is tailored to the traditional environment. One would typically see formulations like: one must submit an identification document to someone for inspection (see, e.g., Art. 435f Dutch Criminal Code and Art. 15.4 Wet op het hoger onderwijs en wetenschappelijk onderzoek). For biometrical verification, it is, however, also necessary to provide a bodily feature, but the said provisions do not establish a duty to do so. It seems therefore that even if the proposed amendment to the Paspoortwet is enacted, no statutory duty for co-operation to biometric identification 'at a distance' exists.

3.2.1.2 Identification duties during contract formation

Disclosure of identity during contract formation

When concluding contracts, it is generally not necessary that the contracting parties know each other's real identity. For example, customers buying goods via vending machines do not always know from whom they are buying, and vice versa. However, when problems arise, the customer may want to track down the seller (Grijpink and Prins, 2001).

Formal requirements during contract formation

It is the question whether formal requirements during contract formation demand that one's identity must be disclosed. In principle, the formation of a contract can take place in any form, i.e., offer and acceptance (see chapter 2) are not bound to certain formal requirements. In civil law countries, the basis for this can be found in the principle that contracts are created by *consensus*: the consensus of both parties alone is enough to create obligations. In some cases, the legislator has made exceptions to this general rule. In these cases, offer and acceptance need to occur in a certain form or need to be accompanied by a formality (Hartkamp and Tillema, 1995; Hartkamp, 2001).

Current formal requirements

As a civil law system, the Dutch legal system proceeds from the principle that contract formation can take place in any form. As an exception to this general rule, the Dutch Civil Code (DCC) demands that certain contracts require a formality in order to be concluded validly; Article 3:39 DCC states: 'Unless the law produces a different result, juridical acts which have not been performed in the prescribed form are null.' Two form requirements can be distinguished: first, the requirement that a (condition of a) contract needs to be in writing. Secondly, in some cases the formation of a contract requires a signed document, an *instrument.* There are two types of instruments: notarial instruments and private instruments. According to Article 156 of the Dutch Code on Civil Procedure, an instrument is a signed document, intended to serve as evidence. A notarial instrument is an instrument that is drafted by a public or civil servant in accordance with certain requirements. Private instruments are instruments that are not notarial instruments. Two examples of situations where Dutch law demand a private instrument are contracts for hire-purchase and collective labour contracts.

Instruments do not allow for anonymity or pseudonymity: the signature on the document should relate to an identifiable person.

European Directives

Formal requirements might hamper the development and growth of electronic commerce. Therefore, Article 9 of the European directive on electronic commerce requires the EU member states to ensure that their legal systems allows contracts to be concluded by electronic means. Article 9 states explicitly that the member states must ensure that the legal requirements applicable to the contractual process neither creates obstacles for the use of electronic contracts nor result in such contracts being deprived of legal effectiveness and validity on account of their having been made by electronic means.

Electronic formal requirements

Because of Article 9 of the Directive on Electronic Commerce, EU member states will have to adapt their formal requirements if they obstruct electronic contracting. This can lead to the formulation of 'electronic formal requirements'; i.e., formal requirements that can be met whilst contracting electronically. These new electronic formal requirements explicitly demand that the true identity of the contracting parties must be revealed.

Article 9 of the Directive on Electronic Commerce has not yet been implemented into the Dutch legal system. However, there is a draft act ; the so-called Adaptation Act. The Adaptation Act introduces a new kind of electronic formal requirement. According to the new Article 6:227a of the Act, when contracting electronically, a statutory requirement that a contract must be formed in *writing*, is fulfilled, if the following requirements are met:

- the contract must be accessible for the parties;
- the authenticity of the contract must be sufficiently safeguarded;
- the moment of the conclusion of the contract must be determinable to a sufficient degree;
- the identity of the contracting parties must be determinable to a sufficient degree.

It is clear that this provision does not allow for anonymity. The answer to the question whether it allows for pseudonymity, depends on the interpretation of the words 'to a sufficient degree'. According to the explanatory notes that accompany the Adaptation Act, Article 6:227a should have the effect that electronic and written contracts are treated the same. This should be accomplished, according to the explanatory notes, by interpreting the vague terms such as 'to a sufficient degree' in such a way that the same result as a written contract is reached. Therefore, the electronic formal requirements demand the same identifiability as the written formal requirements do. One method of 'identifying oneself to a sufficient degree' is by using an electronic signature, the explanatory notes state.

The German implementation of Article 9 of the directive has led to two new formal modalities: the so-called electronic form and the text form, Articles 126a and 126b of the German Civil Code respectively. When there is a formal requirement that demands that a written form must be used, then under the following conditions, this requirement can be met by the electronic form:

- the declaring party must add his name to the electronic document;
- he must sign the document with a qualified electronic signature;
- in case parties want to conclude a contract, both parties must sign an identical copy of the document with a qualified electronic signature.

In case there is a formal requirement that demands that text form must be used, the following requirements must be met:

- an instrument or other durable reproduction of the statement must be provided;
- the statement must be made in letters;
- the declaring party must add his name;
- the conclusion of the statement must be marked by a reproduction of the name of the declaring party or must be made recognisable in an other way.

Apart from the Directive 2000/31/EC, also Directive 97/7/EC has a bearing on identification duties.

Relevance of Directive 97/7/EC on distance selling

The directive 97/7/EC is applicable to certain contractual arrangements. The arrangements, covered by the directive, have to meet four conditions:

1. the supplier makes exclusive use of one or more means of distance communication,
2. the aspect of distance is only relevant in the period up to and including the conclusion of the contract,
3. the directive only addresses relations between suppliers and consumers (B2B relations are excluded) and
4. the supplier uses an organised distance sale or service-provision scheme.

According to the fourth article of the directive, the supplier must provide his identity and, in case of contracts requiring payment in advance, his address. This (and other information) must be provided in good time prior to the conclusion of any distance contract. The consumer must receive confirmation of this information in writing or in another durable medium available and accessible to him, in good time during the performance of the contract or at the latest at the time of delivery (where goods not for delivery to third parties are concerned). The confirmation can be left behind, if the information has already been given to the consumer prior to conclusion of the contract in writing or on another durable medium available and accessible to him (art. 5 Directive 97/7/EC). A 'durable medium' can be defined as 'any instrument enabling the consumer to store information addressed personally and specifically to him and which is mainly contained on floppy disks, CD-ROMs, and the hard drive

of the consumer's computer on which electronic mail is stored.' This means that storage in a mobile agent may not be sufficient, especially if the user has no direct access to the agent.

As discussed above, non-fulfilment of the formal requirement of a writing usually has as its consequence the nullity of the contract in respect of which the form requirement was ignored. In this directive form requirements have been formulated that have an altogether other effect in case of non-observance. If the supplier does not meet the concerned form requirements, the term during which the consumer can dissolve the contract becomes much longer (3 months instead of 7 days). Thus, the directive and its implementations in national law provide an effective incentive to comply with the requirements.

3.2.1.3 Legal requirements

The user of a software agent must adhere to the following duties with respect to self-identification:

- Passive identification duty on the basis of the Dutch Identification Duty Act: this Act requires an identification by means of one of the prescribed means of identification, such as a passport. For the time being, the prescribed means of identification do not lend themselves for on line use. Also, the duties are in need of some reformulation with a view to on line use.
- The active identification duty of art. 5 Directive 2001/31/EC (about the service provider): identificatory information must be permanently available; a website is an adequate means to make the information 'permanently' known. A single message to somebody's software agent is not conform the permanency requirement.
- The active identification duty of art. 6 Directive 2001/31/EC (on commercial communication (a.k.a. spam)): identificatory information must be clearly indicated or referred to in the (spam)message; this requirement does not seem to be problematic with respect to agents, even if the (spam)message is delivered to somebody's agent.
- The form requirement of a writing may no longer be a hindrance to e-commerce (art. 9 Directive 2001/31/EC): in Dutch law the electronic equivalent of a writing must be such that the identity of the contracting parties is determinable to a sufficient degree. This means that a software agent used to conclude contracts must be able to reliably pass on the identity of its user (i.e., one of the contracting parties), so that this can be 'incorporated' in the contract. This may mean that the agent must be able to use the electronic signature of its user.
- Identification on the basis of Directive 97/7/EC (on distance selling): the supplier must make the identification data available to the consumer in writing or in another durable medium available and accessible to him. This means that delivery to somebody's mobile agent is not enough. After all, an agent may – mobile as it is – not be accessible at all times by its user. Furthermore, it seems difficult to guarantee storage on a durable medium.

The producer of a software agent for participating in the on line society:

- Although the producer is not under a direct duty to build an agent that has some or all of the identificatory capabilities described above, he may be under a duty to warn the user if certain identificatory capabilities are lacking. This is especially the case if the user may expect the capabilities based on, e.g., the nature of the agent.

3.2.2 Technical Perspective

The subject of self-identification plays a double role for agents: agents themselves usually have an identity, and an agent may carry confidential information which can be used to identify one or more humans. This technical perspective briefly discusses agent and human identities, techniques for identification, and measures for identity protection.

The example chemical commodity marketplace needs to resolve these issues and publish its approach to facilitate development of compliant agents. The marketplace may provide agents of its own, thereby guaranteeing adherence to its protocols. Techniques and measures described below may apply to the marketplace: choices have to be made, involving reliability, risk, and trust, perceived by the marketplace itself and its (potential) customers.

3.2.2.1 Agent identities and human identities

Agents commonly have some form of identity, for the purpose of distinguishing between agents. This identity does not have to be usable for contacting purposes, such as sending messages: agents can communicate without sending messages to each other, e.g., by writing messages on publicly readable websites. Agents with identities

usable for communication purposes, commonly employ location services in which agents or agent platforms can lookup the agent's contact address (i.e., the Internet location where the agent receives messages, the mailbox), based on the agent's identity.

A possible scheme for using agent identities, relies on each agent having a globally unique identifier, which is hidden in specific location services and agent platforms. Location services and agent platforms only provide 'agent handles', which contain a name of an agent and contacting information. Agent handles may change over time (e.g., when an agent migrates to a new location, or when an agent assumes another name). Different location services may employ different globally unique identifiers for agents.

Agents may publish information about themselves and their users in directory services, with the aim of being addressable by other agents. Whether agents are to provide confidential information, such as personal information and identification information such as banking information, credit cards, money, information about its organisation, information about its user, logins and passwords, and information about the agent's current user, remains the question. A similar argumentation holds for agent platforms (which have identities and may store confidential information about its owners and agents hosted, etc.). Legal duties may require to safeguard *any* confidential information from unwanted disclosure and traceability, including information which may identify humans. Techniques for identification and security measures are discussed in the next subsections.

3.2.2.2 Techniques for identification

In computer systems, an approach is needed to tightly couple identity to an agent or a human. This involves using protocols for authorisation and access control, which in turn employ techniques such as digital signatures, watermarking, certificates, split key, which are often based on cryptographic techniques. Trusted third parties are often included in such protocols. Below, a number of well-known techniques are briefly described, based on (Anderson, 2001; Tanenbaum and Steen, 2002).

- *Encrypt & decrypt*: The main principle in cryptography is that information (termed *plaintext*) can be encrypted with a *key*, which results in unintelligible information (termed *cyphertext*). This unintelligible information can only be restored to its original form when it is decrypted with the same key. The strength of a cryptographic algorithm depends on the mathematical principles of the algorithm, and the choice of the key. Cryptographic systems may differ in their use of keys: *symmetric cryptosystems* use the same key for encryption and decryption; *asymmetric cryptosystems* use different keys which together form a unique pair (see below). In symmetric cryptosystems, both the sender and receiver of a message have to share the same, secret key.
- *Public Key Infrastructure (PKI)*: The basis for PKI are asymmetric cryptosystems in which unique pairs of keys are used for encryption and decryption. The effect is that information encrypted with the public key, can only be decrypted via the private key, and *vice versa*. The encryption key is commonly made *public*, the decryption key is kept *private*, the other combination may also be used. For example, sending a message to Alice first involves encrypting information with Alice's public key and then sending the encrypted information to Alice. Only Alice can decrypt the encrypted information contained in the message, as only Alice has the private key belonging to the public key she handed out. Alice may use multiple key pairs at the same time. Alternatively, Bob may encrypt the information intended for Alice with his private encryption key, and send it to Alice. Alice can then decrypt Bob's message by using Bob's public decryption key. The difference in deciding which key is public, is similar to difference in sending and receiving semantics with respect to message delivery. PKI is a commonly used architecture for encryption and decryption, e.g., in PGP (Anderson, 2001).
- *Digital signatures*: Securely transmitting information does not prevent a party from altering the information received. A digital signature can be used to verify the integrity and source of the information. A digital signature is often based on a *hash-function*, which is an irreversible mathematical function which computes a unique number, based on information (plaintext). The originator of the information computes the hash value of the information and encrypts it with its private encryption key: this is a digital signature. The information together with the digital signature are sent to the receiver (in a secure way). The receiver can decrypt the digital signature (with the sender's public decryption key), which results in the sender's hash value. The receiver then compares the sender's hash value with the results of applying again the hash function. If the recomputed has-value differs from the sender's has value, then the information has been tampered with (Anderson, 2001; Tanenbaum and Steen, 2002).
- *Split key*: Providing agents with private keys may entail a security risk, as they may be compromised. A solution to this problem is to provide an agent with part of the key. The other part of the key is sent by the user to the agent when it is needed, and subsequently forgotten (Shamir, 1979).
- *Cloaking and Watermarking*: Another solution to providing agents with private keys, is to hide the private key: *steganography*. The principle is as follows: the originator wishes to hide information in some other

cover-information (text, images, code, audio, video, …). To this end a transformation process is used, which uses a *stego-key* and produces *stego-information*. The latter is openly readable and useable by other people; the embedded information, however, can only be extracted by employing the right stego-key and the inverse of the transformation process used. Cloaking entails hiding a key in the code of an agent (e.g., Nickerson, Chow and Johnson, 2001), while watermarking entails hiding identificatory (e.g., the copyright owner) information in images, audio, and video (Miller, Cox, Linnartz and Kalker, 1999).

- *Certificates*: A certificate is basically information which is digitally signed by the certification authority: a trusted third party (Anderson, 2001). Techniques such as described above can be employed. Certificates may be used as alternatives to providing identificatory information: e.g., when an agent can hand over a certificate which specifies that the agent is allowed to do a certain action, the identity of the user or owner of the agent need not be disclosed.

A common issue related to cryptographic techniques involves distributing keys. One needs to verify that a specific key belongs to a specific entity (and not a pretender). Solutions exist but differ in their scalability, flexibility and consistency, a common difference between decentralised and centralistic approaches. For example, an agent may consult known entities ('webs of trust') or inspect trusted public databases containing public keys of agents.

A number of issues play a role in protecting confidential information and agent identities:

- *Mobility*: software agents may migrate to locations, some of which may be malicious or non reliable for other reasons. Confidential information in mobile agents has always a risk of being disclosed unless encrypted.
- *Cloning*: software agents may be cloned, i.e., a copy which is completely the same (including confidential information). It is debatable whether a clone should have the same identity as the original, or a different identity. However, in some definitions of cloning the clone is always the same as the original, to the extent that any information given to a clone is also known to the original (note that some agent platforms may not support multiple agents with the same identity). In other cases, when the clone of an agent "runs its own life", a different identity is assigned. This issue requires more research.
- *Aggregation*: software agents that are grouped together, may also have a collective identity. Examples include all agents of one user or agents that currently work on a shared problem. This collective identity may be used for communication purposes, but also, perhaps, for acting. In the latter case, usually a specific agent assumes the collective identity and is able to act. This issue also requires more research.

The environment of software agents also plays a role in protecting the agent's confidential information:

- *Protecting an agent from agent platforms*: a mobile agent may reside at numerous hosts. in general, an agent implicitly has some trust in the agent platform on which it runs: a computer has complete control over all agents it hosts, but not necessarily all information known to agents (some may be encrypted). Although tracing techniques may be used to detect whether an agent platform disobeys protocols, damages may still occur.
- *Protecting an agent from another agent*: Other agents may also disobey protocols, possibly causing damages.
- *Protecting an agent from other software (objects, services, …)*: Agents interacting with objects and services may run a risk when an object or service disobeys protocol.
- *Protecting an agent platform from agents*: Agent platforms use access control policies to decide which agents to host. Such access control policies may favour agents which have an identity of their own (i.e., can be traced), and, e.g., whether the identity of their human designer, human owner, and/or human user is known or traceable (e.g., via trusted-third parties). This may conflict with the needs of some human users, who may wish to remain anonymous at all times. Safe protocols are required, which is subject of further research.

Designing security techniques is extremely complex. A very important remark is that cryptographic *algorithms*, which form the basis of security techniques, only provide temporary security. Advances in mathematical theories and computing hardware will almost certainly 'break' the algorithms in the future. It is argued that it is difficult, or even impossible, to build secure systems (e.g., Anderson, 2001; Tanenbaum and van Steen, 2002).

A side remark about biometrics (e.g., using fingerprints or retina-patterns as a key in algorithms) is that biometrics is *not* a panacea: the digital information about the pattern may be compromised. As a result, biometrical data may become obsolete. Humans cannot easily obtain new fingerprints or retinas to use as digital keys.

In general, it can be concluded that:

- techniques for identity protection are not failsafe: minimal confidential information should be placed in software agents and agents should often change their keys, and use debugged cryptographic systems.

- it remains difficult to verify whether the agent's information about its user actually identifies its user.
- all security measures are of little value if another agent publicly announces the identity of the user together with the identity of the software agent. It then becomes possible for agents previously met, to retroactively deduce the user of the agent.

3.2.2.3 Measures for identity protection

Based on the current technological situation options are to

- place minimal confidential information in a software agent (and obtain more confidential information if required).
- use appropriate techniques to hide confidential information.
- use appropriate protocols when interacting with other agents and agent platforms.
- place appropriate protective strategies in a software agent.
- reflect on the consequences when confidential information becomes (semi-)public.
- use appropriate access control policies in agent platforms.

Research is needed to determine what 'appropriate' entails: to this end protocols and techniques need to be analysed and tested to verify their robustness and reliability in terms of computational expense, temporality, and legality. An approach may be to relate 'appropriateness' to the tasks an agent performs.

The external behaviour of the agent, denoted as 'tasks', can be taken as an indicator for the amount of confidential information stored in the agent, and techniques and protocols needed. For example: information retrieval, decision support, and transaction making.

- Information retrieval: an agent retrieves information from sources on the Internet, including other agents. The agent only needs sufficient identificatory information, techniques and protocols, to gain access to information. In some cases, a user may be anonymous, in others the user may need to explicitly acquire access.
- Decision support: an agent advises users and/or other agents about possible actions, e.g., transactions. The agent only needs sufficient identificatory information, techniques and protocols to interact with users and/or other agents. This task involves more extensive interaction between agents and may require more extensive information, to gather information needed for a decision (including, e.g., draft contracts) and to provide credibility to offered advice.
- Transaction making: an agent is directly involved in a transaction. The agent needs sufficient identificatory information, techniques and protocols to finalise a transaction, on behalf of its user(s) with specific parties. This task requires more extensive information, which should be kept minimally sufficient for interaction with expected parties.

Agents often are involved in several (external) tasks (not necessarily at the same moment in time). In this case, an agent needs identificatory information, techniques and protocols for the most 'complex' task. The trade off between the amount of identificatory information and the choice of techniques and protocols depends on mutual interest and risk. Whenever the mutual interest of users and other agents becomes greater, stronger forms of identificatory information may be required. When the perceived risk increases, stronger techniques and more stringent protocols may be required.

Possible approaches to avoid using a software agent with confidential information:

- send agents without any confidential information to report back with information upon which the user can take action, e.g., an agent searches for a book in a bookstore, while the human user buys the book.
- use (e.g.) electronic cash which gives legitimacy to a specific action to be undertaken by an agent and which does not require information about a human, e.g., an agent searches for, and buys, a digital book and brings it to the human user.

When agents have the ability to dynamically change the amount of identificatory information carried, agents may initially have minimal identificatory information, acquire necessary information for a specific purpose, and afterwards 'forget' this information.

3.3 RECIPE FOR ANONYMITY

Sometimes a person wants to hide his or her identity or participate under a pseudonym in the societal life. The bidders at an auction may, e.g., want to hide their identity in order to avoid a negative influence on the price-formation. A person may thereto make use of a software agent that hides his 'true' identity.

3.3.1 Legal Analysis

In order to discuss anonymity, one has first to know or at least delimit what anonymity is. This is, however, a far from easy task, since anonymity is as of yet not an established legal concept. In this section, the legal concept of anonymity is explored to some extent.

3.3.1.1 Definition issues

Anonymity is in the first place characterised by the fact that other parties do not know ones identity. Ones 'identity' is defined as ones name or other human readable data under which a person is known. This is a weak concept of anonymity, since the absence of knowledge about ones identity does not imply that the identity cannot be found out, nor does it imply that there may be other ways to access a person than via his identity. In other words, it does not imply untraceability. The strong notion of anonymity does involve untraceability. Determining whether there is anonymity (in the strong notion) is not completely straightforward, as untraceability requires an assessment of a factual situation: the traceability of a person. Absolute untraceability (untraceable in no situation whatsoever) may be no more than an ideal. So untraceability is rather a relative notion: one is untraceable if the cost of tracing (largely) exceeds the expected gains to be won by a successful trace.

3.3.1.2 Legal status of anonymity

As of yet, there does not exist a right to anonymity, although a person is in principle not prohibited to try and find anonymity with the help of organisational, technical or contractual means. A software agent hiding the identity of its user while acting on the Internet is therefore in principle allowed.

This also holds for contracting. The key principle of the Dutch contract law is that contracts can, in principle, be entered into without prescribed form: 'unless stipulated to the contrary, declarations, including notifications, can be given in any form and can be incorporated in behaviour', reads Article 3:37, paragraph 1, of the Dutch Civil Code. Unless opposed by imperative law, the parties are free to incorporate in the contract the obligation that their mutual identity is specified based on the principle is that the parties themselves determine the method used to declare their intent. This could, therefore, be an absolutely anonymous one. This makes absolutely anonymous electronic legal transactions possible. Thus, it also allows for the use of agents that do not reveal the identity of its user.

3.3.1.3 Limitations of anonymity

Anonymity is limited in that a person cannot deny identification duties that rest upon him (see for identification duties the previous recipe). The mere existence of an identification duty does, however, not mean that there is no anonymity. The anonymity is only lifted by observance to the identification duty. Someone wishing to protect his anonymity may therefore want to evade situations in which such a duty must be fulfilled.

As noted above, a person may in principle seek anonymity using the means he sees fit. The reverse does, however, also hold: other people may, in principle, try to find out the identity of someone who is anonymous. Someone trying to unveil the identity of an anonymous person must, however, observe the law in doing so: he may, e.g., not infringe upon the privacy of the anonymous person, he is not allowed to hack into computers or wiretap telecommunications. A grocer may, e.g., not just like that have video cameras in his shop filming his customers. He will have to abide to privacy rules, that, e.g., oblige him to have a notice at the door, warning the customers that they may be filmed.

From these examples it appears that a number of legal rules exist that can be helpful in protecting ones anonymity, although 'anonymity' is not the prime object ("rechtsgoed") that is protected by the rules. One could say that those rules provide 'flanking' protection to anonymity. Any acts aimed at finding out a person's identity that are not unlawful may thus be used. One may, e.g., ask a third person to disclose the identity of an anonymous person. The third person is, of course, under no duty to disclose the identity. In certain circumstances, the law provides a means to oblige the third person to answer:

> In case somebody wants to institute legal proceedings against a person whose identity he does not know, he may request a court to grant a provisional examination of a witness (art. 214 Dutch Code of Civil Procedure; hereinafter: DCCivP). The witness that is to be heard, is of course somebody who is thought to have knowledge of the identity of the person against whom legal proceedings are to be instituted. From Dutch case law, a case is known in which an ISP has been summoned to appear as a witness in order to disclose the identity of one of his subscribers to a copyright holder, who accused the anonymous subscriber of copyright infringement [Rechtbank Arnhem 15 juni

1999 LJN AA1015]. A witness has a duty to answer the questions that are put before him, including questions about identities.

A special case arises if it is the government or a government agency that seeks to lift ones anonymity. For an agency to be authorised to do so, the law must have vested a power in the government agency to take the measures, that it envisages to use in unveiling the identity. Such a power may not only authorise the agency to take identification measures with respect to a person, it may also be a justification to take measures that infringe upon 'flanking protection'.

> According to art. 61a Dutch Code of Criminal Procedure (hereinafter: DCCP), a suspect that has been arrested and has been brought to a place for interrogation (typically a police station) may be subjected to identification measures if such measures are necessary to determine his identity. The admissible identification measures are: photographing the suspect, taking his fingerprints and measuring his body.
>
> An investigating officer that has stopped or arrested a suspect may search the suspect's clothing, and the objects that the suspect carries with him, if such is necessary to determine his identity (art. 61c DCCP). In certain circumstances, this power may even be exercised in a public place.

In the context of criminal investigations no power specifically aimed at identifying persons that take part in online communications exists. However, a public prosecutor may (in certain circumstances)[4] order telecommunications operators (including ISPs) to inform him about all traffic that has been handled by the network or service and which the suspect is presumed to have taken part in (art. 126n and 126u DCCP). This power is more commonly known as the power to order 'traffic data' to be supplied. The traffic data that are supplied following such order, may identify the suspect or make a subsequent identification easier. The former may be the case if the operator supplies the name of a subscriber to his service, the latter may be the case if data about the whereabouts of a cellular phone are supplied.

3.3.1.4 Legal protection of anonymity

Legal protection of anonymity seems to be at odds with the absence of a right to anonymity, as noted above. One could be forgiven to think that anonymity can only be protected by organisational, technical and contractual means. Nonetheless, the law may in circumstances also provide protection. Although anonymity is as such not protected in general, this does not mean that anonymity cannot profit from protection under flanking legal concepts. The following examples of flanking legal protection exist: informational privacy, the penalisation of hacking, the confidentiality of telecommunications, and the right to respect for ones home.

With respect to software agents the most pressing problem with respect to anonymity may be the following. A software agent contains information that identifies or could help to identify its user, which information it needs to fulfil its function or task. Nonetheless, the agent is programmed to maintain the anonymity of its user. Since it is also a mobile agent, it finds itself often on computer systems 'far away from home' that may not be sympathetic to the idea of maintaining a user's privacy. The question thus is the following: are the data in a software agent legally protected against inspection by the owner of the system on which the agent finds itself? If so, under what conditions? In this respect, the Dutch computer crime II bill is relevant [Tweede Kamer 1998/99, 26671, nrs. 1-2]. In this bill the following provision (Art. 273d Dutch Criminal Code) can be found:

> With imprisonment of at most one and a half year or a fine of the fourth category (i.e., Euro 11250,-) will be punished a person employed by a provider of a public telecommunications network or –service:
>
> a. who wilfully and without right inspects data that have been stored or are processed or transmitted with the help of such network or service and that are not destined for him, or who wilfully and without right copies, taps or records such data for himself or another.
> b. […]

If and when this provision will be enacted, data inside a software agent are protected against unwanted inspection. Two of the conditions that must be met, are dealt with more intensively here: what does 'without right' mean? What is a provider of a public telecommunications network or –service?

Without right: according to the explanatory memorandum this addresses the situation that a provider of telecommunication without authorization of the person concerned inspects personal data of his customers that

[4] These circumstances comprise the following: The suspect has been caught red-handed, or there exists a suspicion that a severe criminal offence (usually punishable with four years or more) has been committed or there exists a suspicion of hacking.

are stored in his computers (e.g., e-mail in an e-mailbox) [Tweede kamer 1998/99, 26671, nr. 3, p. 47]. So the protection only seems to extend to 'personal data'. The example mentioned (e-mail) further seems to highlight that the personal data must be recognisable as such. In case of a mailbox this is clear: a mailbox is recognisable as such and everybody expects to find personal data in a mailbox. That a software agent contains personal data is, however, not self-evident. So if an agent contains personal – especially identificatory – information it is relevant that the information is recognisable as such and cannot be inspected 'by accident'. So, in order to qualify for legal protection under this provision, it is necessary to store these data in encrypted form. However, it is unlikely that the encryption needs to be especially elaborate. Probably, it suffices to use a rather simple form of encryption.

The second condition that has to be borne in mind is that the provision is only applicable to providers of public telecommunication networks or –services. Telecommunication is not to be interpreted in a narrow way: not just traditional telecommunications operator (e.g., KPN) but all telecommunications providers are covered by the term: thus also, e.g., Internet Access Providers. However, the networks or services they provide must be public, meaning it must be available to the public. In closed networks, the provision is not applicable. It is then up to the parties to agree on the confidentiality of the data in agents, if they desire to do so.

3.3.1.5 Towards a right to anonymity?

The Commission 'Grondrechten in het digitale tijdperk' advised the Dutch government against the adoption of a general right of anonymity in the Dutch Constitution (See the Commission's report on: http://www.minbzk.nl/gdt/index2.htm). The reasoning was that the number of exceptions to the right that would have to be allowed would be so large, that the right would become virtually meaningless. The Dutch government concurred with the commission on this topic and no proposal to amend the Constitution in this respect has materialised [Tweede Kamer 2000/01, 27460, nr. 1]. A drawback of the position is that the symbolic value that such an adoption would have had is not to be. A symbolic function can be very important in drawing attention to the right of anonymity, and in fostering support for such a right.

A concrete result of the adoption of a right to anonymity could have been a restriction of the freedom of contract: in certain circumstances a party would not have been allowed to refrain from contracting, only because the other party does not divulge its identity.

3.3.1.6 Legal requirements

The user of a software agent containing identification information:

- If the user's software agent contains data that identify or can identify the user, these data must be encrypted if the user wants to remain anonymous. Without such encryption, it will be difficult to claim legal protection for the anonymity. The assumption used here is that Art. 273d DCC will be enacted conform the proposal CCII.
- If the agent containing identification information functions in a closed network, legal protection for the anonymity of the user has to be sought in contractual arrangements. In this case, it is perhaps possible agree upon a marker for confidential information; then encryption may, at least from a legal perspective, no longer be absolutely necessary within the closed network.

The owner of a system on which mobile agents reside:

- The owner of a public telecommunications network or service may not – without authorisation - inspect identificatory information in a mobile agent and that evidently must remain confidential. In practise, this probably means he may not try to decrypt encrypted data.
- In cases a framework contract for the use of a closed network exists in which parties have agreed upon markers for confidential data, the system owner may not seek access to data that are present in a mobile agent and that are marked as confidential. The system could perhaps be programmed to recognise the markers and give no or no unclausulated access to the data so marked.

3.3.2 Technical Analysis

Anonymity may apply to users, agents and even agent platforms (e.g., to facilitate agents's anonymity or pseudonymity). These are briefly discussed in this section. For example, in the grocery shopping scenario, the user of a shopping cart wishes to remain anonymous until he or she buys products. Similarly, an agent acting on behalf of the user, which visits the store, wishes to remain anonymous until products are bought. In both instances pseudonymity may be more useful, as the shopping cart is to be used by one user (or agent).

Anonymous agent platforms are less likely to be used in the grocery shopping scenario, as all agents involved need assurances from their hosting agent platforms: assurances are commonly tied to an identity.

3.3.2.1 User anonymity

Anonymity of human users involves hiding identificatory information in an agent. See Section 3.2.2 for a description of techniques to hide identificatory information. The agent containing hidden identificatory information needs to understand protocols via which identification duties may be enforced. The agent needs a secure mechanism to disclose identificatory information to the right party.

3.3.2.2 Agent anonymity

If total anonymity needs to be achieved, i.e., an agent needs to be truly anonymous, cannot be recognised nor traced at least the following restrictions are to be placed on an agent:

- The agent's body does not have any discerning features.
- The agent's identity is unknown to other agents/users.
- The agent's information which may identify its user(s), may be encrypted.
- The agent's actions are untraceable.
- The agent's communication is untraceable (e.g., no traffic analysis).

These restrictions are either improbably or require elaborate mechanisms:

- An agent's code and data are usually unique, but techniques may be applied to hide differences (e.g., using standard code, hiding an agent's private data in a remote object, etc.).
- In most agent platforms an agent needs to have an agent identity to participate in the system. An approach may be to often change the agent's identity (if supported by an agent platform), which may hamper the agent's migration, communication and interaction abilities.
- An agent carrying identificatory information needs to hide this information, see Section 3.2.2 for a discussion of a number of techniques.
- An agent commonly leaves a trail by communicating, acting, migrating, and by being included in location services, etc… Hiding these trails cannot be easily achieved because of the distributed nature of the trail; obfuscation may be a useful approach (see Section 3.4 for services facilitating anonymity and pseudo-anonymity) as agent platforms autonomously build traces and cannot be expected to make exceptions for anonymous agents.
- An agent's communication is always traceable in some form, especially in point-to-point messaging, allowing for traffic analysis. Even when an agent posts notices on bulletin boards, some identificatory information is used (e.g., code phrases, encoded messages, or other distinguishing marks) so that communication participants can recognise relevant messages.

Such an extreme form of anonymity severely restricts the usability of an agent:

- The agent may not be hosted by an agent platform (an agent which does not 'run', is not very usable).
- The agent may not be able to migrate to other agent platforms.
- The agent may not be allowed to access certain information.
- The agent may not be allowed to communicate with other agents.
- The agent may not be allowed to participate in certain actions, including altering information, transacting, creating or deleting information or agents, etc.

Agent anonymity in its extreme form is not easily achieved, and may probably be impossible because of restrictions imposed by most agent platforms (including the need for globally unique agent identifiers). However, some forms of anonymity and pseudo-anonymity may be feasible, see Section 3.4. In any case, it is safest to assume that absolute anonymity is never the case, temporary anonymity may be realisable.

3.3.2.3 Agent platform anonymity

An agent platform may also be anonymous or employ pseudonyms, e.g., to offer anonymity and pseudonymity services to agents. Anonymity or pseudonymity of agent platforms may severely restrict the useability of an agent platform:

- location and directory services may contain incomplete and inaccurate information about an agent platform,
- other agent platforms may negatively adjust their assessment of an anonymous or pseudonymous agent platform and as a consequence block migration and even communication with agents on such an agent platform,

- agents may refuse to go migrate to an anonymous or pseudonymous agent platform as it may be unsecure, unreliable, etc. (hard to enforce legal obligations),
- agents hosted by an anonymous or pseudonymous agent platform may also be less usable: agents may not be able to migrate to other agent platforms; and their communication abilities with agents on other agent platforms may be hampered.

3.4 FACILITATING ANONYMITY AND PSEUDONYMITY

With respect to software agents the most pressing problem with respect to anonymity may very well be the following. A software agent contains information that identifies or could help to identify its user, which information it needs to fulfil its task. Nonetheless, the agent is programmed to maintain the anonymity of its user. If the agent is also mobile, it may often reside on computer systems 'far from home'. These systems may not be sympathetic to the idea of maintaining a user's privacy. The question thus is the following: are the data in a software agent legally protected against inspection by the owner of the system on which the agent finds itself? If so, under what conditions?

3.4.1 Legal Aspects

This recipe deals with persons acting anonymously or using a pseudonym. The subject of this paragraph of the recipe is the position of the party that is offering services that allow others to act anonymously or under a pseudonym. The following questions will be addressed:

- When is it unlawful to offer services that allow others to act anonymously or using a pseudonym?
- When must providers of services that allow others to act anonymously or using a pseudonym, provide the means to remove the anonymity and pseudonymity of these others?
- Should persons acting anonymously or using a pseudonym be notified that their true identity can be traced?

These questions will be answered according to Dutch law. For the sake of convenience, the person that is providing the anonymity- or pseudonymity services will be referred to as the anonymity provider.

3.4.1.1 When is it unlawful to offer services that allow others to act anonymously or using a pseudonym?

Anonymity and pseudonymity can allow persons to perform acts, without others being able to see what the real identity is of the persons performing these acts. Therefore, it can be possible to act criminally or unlawfully without others, including law enforcement authorities, to find out the identity of the perpetrator. The relevant criminal acts might be (amongst others):

- The distribution or making available and/or the possession of child pornography;
- The incitement to discrimination;
- The making of discriminatory remarks;
- Acts of libel or defamation (Roos and Wissink, 1996).

The anonymity provider is not the one performing these criminal acts, but he might be viewed as a participant in the criminal activity. In criminal law, two degrees of participation can be distinguished: *co-authorship* of the criminal act and being an *accessory* to the criminal act (Toorenburg, 1998).

Co-authorship occurs if two requirements are fulfilled: (1) A conscious co-operation between the two (or more) perpetrators, and (2) a joint execution of the act. It is obvious that an anonymity provider can be criminally liable if he consciously works together with the (other) perpetrator in performing specific criminal acts. Therefore, under these rather extreme circumstances, it is illegal to offer services that allow others to act anonymously or under a pseudonym.

One can be considered an accessory if (1) one is intentionally helpful in committing a criminal offence, or (2) one intentionally offers opportunity, means or information for committing a criminal offence (Article 48 Dutch Penal Code). The requirement under (2) might offer problems for anonymity providers. Does an anonymity provider offer opportunity or means to commit a criminal offence? The *intent* of the anonymity provider is crucial. So-called conditional intent is sufficient, i.e., it is sufficient that somebody accepts a chance or a risk that need not be bigger than that the possibility of its materialisation cannot be discarded as imaginary. Does this make an anonymity provider liable for criminal offences that afterwards appear to have been committed with the help of the services of the anonymity provider? Probably not, the mere fact that a service can be misused does not amount to conditional intent of the service provider. If, however, the service provider knows (or with a high degree of certainty suspects) that one of his subscribers uses his anonymising services for criminal purposes, while he can very well prevent such, he *must* take measures to prevent the misuse. Failing to do so, may amount

to complicity to the crime of the subscriber. As a rule of thumb, one can say that the legal risks for the provider are smaller if he merely supplies software (e.g., an anonymising software agent) than is the case if he runs an anonymising server. In the latter case, the capability of the provider to act is much greater. In the former case, the provider only supplies the software and once it is out of his hand, he generally cannot exert influence anymore.

3.4.1.2 When must providers of services that allow others to act anonymously or using a pseudonym, provide the means to remove the anonymity and pseudonymity of these others?

If an anonymity provider fits the legal definition of 'telecommunications provider', then certain requirements that a telecommunications providers has to meet, must also be met by the anonymity provider. Some of these requirements relate directly to the identify-ability of the users of the anonymity- and pseudonymity services. According to the Dutch Telecommunications Act, a telecommunications provider may only offer his services if his network can be tapped. Subsequently, law enforcement authorities can demand a tap of a certain communication. Such a tap may entail the identification of the sender and the recipient of the communication.

According to Dutch caselaw, for the assumption of a duty to divulge the identity of an anonymous actor two factors are relevant [HR 27 november 1987 BIE 1987, 25 (Chloe/Peeters) and RB 's Gravenhage 9 juni 1999, Informatierecht/AMI 1999, p. 110-115, m.nt. K.J. Koelman]. In the first place, it must be clear that the anonymous actor acts unlawful. It may not be necessary that there is a court order, but the unlawful character must be beyond doubt. In the second place, there must be a risk that the damaging activity will continue, if the identity is not divulged, or there must be a risk that the damage cannot be recouped, if the identity of the anonymous person is not made known. These two factors have to be weighed against the privacy-interest of the anonymous actor. Such a privacy-interest will, e.g., put more weight in the balance if it concerns personal communication that is covered by a (constitutional) telecommunications secret. There is a strong case for e-mail to be covered by the telecommunications secret, e.g., on the basis of art. 8 ECHR. Caselaw explicitly confirming this is to our best knowledge presently lacking.

In general there is no duty to register the identities of persons who make use of the anonymising service. This may, however, be different if the provider knows that the customer will abuse the anonymity service for criminal or otherwise unlawful activities. In such case, the provider must of course disallow the customer to use the service. Allowing such person to use the service may be construed as complicity to the crime of the customer, as discussed above. If he allows the customer to make use of the service, then 'not having registered' the identity of the customer may in such case even further weaken the position of the provider.

A problem may be that the provider of a service only knows an identity that appears afterwards to be false, as was exemplified by the scenario at the beginning of this chapter. In such case the provider may be asked to provide information that holds clues to the identity, such as IP-addresses, telephone numbers etc.

3.4.1.3 Should persons acting anonymously or using a pseudonym be notified that their true identity can be traced?

There is no statutory requirement for anonymity providers to notify their users that their true identity can be traced. However, an anonymity provider might be contractually obliged to inform the users. If the anonymity provider claims to offer true anonymity, but cannot deliver true anonymity, then he might be in breach of contract. However, this depends on the exact circumstances of the case at hand.

Under telecommunications law, the provider of a public telecommunications service must inform its subscribers if there exists a particular safety risk, that a customer normally would not expect [Art. 11.3 Telecommunicatie-wet]. The customer can in such case measures that are appropriate, given the purposes for which he uses the anonymising service.

3.4.1.4 Legal requirements

Two actors are involved:

- The user of a software agent that enables to take part in the 'on line society' without revealing ones identity.
- The provider of the anonymising software agent.
- The producer of the agent.

The user of an anonymizing software agent or an agent making it possible to act pseudonymously:

- If the customer discovers that his identity has become known in spite of the anonymizing service, he must take all measures to prevent further damage. Not doing so may mean that the damage cannot or to a lesser exist recouped from the service provider.

The provider of the anonymizing service:

- The provider must see to it that he does not provide anonymising services to persons of whom he knows or with a high degree of certainty suspects that they will use the service for criminal or otherwise unlawful purposes.
- If there exists a safety risk that may lead or may have led to a revealing of the true identity of an anonymous or pseudonymous person, the provider must warn the user of the service of this risk. The basis for this duty may be found in telecommunications law (in case of public services) or in contract.

The producer of an anonymizing software agent:

- He must warn for safety risks, i.e., risks that the identity becomes known
- He must not supply the agent to a person of whom he knows (or almost certainly suspects) that he will use the agent for criminal or otherwise unlawful acts.

3.4.2 Technical Analysis

Users, providers, and producers of services which facilitate anonymity or pseudo-anonymity are concerned with authorisation and access control, risks, and facilitation techniques.

3.4.2.1 Authorisation and access control

Determining which agent (or user) is to have access to a anonymisation service is difficult. From a legal point of view, only users, or agents acting on behalf of these users, may use these services if the provider or producer does not have a suspicion that a user, of the agent acting on behalf of a user, has an intent to engage in criminal or otherwise unlawful acts. Assessing criminal intent is hard, as it involves disclosing identificatory information, and actually identifying and analysing intent. Protocols and certificates may be employed to achieve some form of certification scheme, but whether it will be waterproof against criminal intent? Analysis of agents cannot offer useful information on possibly criminal intent of an agent (it is impossible to discover any 'intent' inside an agent). Services requiring human users to fully identify themselves, in order to ascertain non-criminal intent or histories, may not be very popular, as the anonymisation service now holds extensive identificatory information about its users. Protocols are needed which specify rights of parties involved, e.g., XrML, a commercial extensible rights mark-up language (http://www.xrml.org).

Services (and their providers) need to provide information to their users (and agents) about their conditions of use, as they may be obliged to use protocols to disclose information about specific users in specific (legal) circumstances. This latter protocol needs to be secure, so that information is securely brought to the intended parties. Information about specific users may include traffic information and identificatory information.

In the chemical commodity marketplace, policies may be in place to regulate which agents are allowed to enter: e.g., agents which are registered at specific third parties. This may inspire trust in whether the agents in the marketplace are, e.g., identifiable. Users who have misbehaved at the marketplace (e.g., failing to deliver goods), may be banned, which implies that their agents are to be banned. However, forcing agents to disclose their user's identity when entering the marketplace is only in violation to the need for anonymity at the marketplace if the marketplace publishes the users of agents, otherwise the marketplace is a trusted third party for all users.

3.4.2.2 Risks & recovery

Users of an anonymisation service need to be aware of the risks involved. Protocols and certificates can be used to indicate risks in using the service, which may include information on:

- conditions of use,
- regulations concerning tracing data,
- techniques used,
- legal conditions for disclosing tracing data and other data.

A first decision about risk (and trust, see Chapter 5), is whether the user (and its agents) trusts a specific service which facilitates anonymity and pseudonymity. When registering to a service, minimal identificatory information is to be provided. Agents using an anonymisation service need to understand these protocols and certificates, to decide whether the service is adequate to their needs.

The computer from which the user uses the service, and the agents communication / acting through the service, may contain information about their anonymity or pseudonymity. When these agents or computers are compromised, external parties may be able to combine information on anonymous agents and discover more information about their users or agents.

Discovery of disclosed identities may not be easy, in a large-scale distributed system. Recovery from disclosed identities is also difficult, especially for human users, whose identity is more-or-less static, although they may assume new pseudonyms. Agents whose anonymity is compromised can more easily be replaced by new, anonymous or pseudonymous, agents.

In the chemical commodity marketplace, a large risk may be associated to prematurely revealing the identity of the user of an agent, e.g., its counter parties may then be better informed about the agent's instructions, including price limits, quantity constraints and temporal constraints, providing them with an advantage during negotiations.

3.4.2.3 Services

Current services facilitating anonymity and pseudonymity are to be used by human users, e.g., for anonymous email and surfing. Below, a number of these services are listed:

- anon.penet.fi was a centralised double blind remailing service until ca. 1996, when it was legally forced to disclose identities of users, after which it was shut down (Martin, 1998).
- www.anonymizer.com offers centralised anonymous surfing, but may disclose information about users during their surfing session or afterwards on the basis of logs (Martin, 1998).
- rewebber.com (formerly JANUS), also offer centralised anonymous surfing, but is designed for anonymous publishing (Martin, 1998; Rieke and Demuth, 2001).
- www.onion-router.net; onion routing is a decentralised approach (based on mixing, used by, e.g., MixMaster), in which a message is transmitted over a number of intermediate nodes, each of which have their own PKI-pairs. The 'onion's are the layered encrypted messages, the 'onion routers' are the forwarding nodes. The last node is able to unpack the message, and send it to its destination. Onion routing obfuscates message content, message origin and destination, and its implementation impedes traffic analysis (Martin, 1998; Goldschlag, Reed and Syverson, 1999).
- Crowds is an alternative to onion routing, more akin to a peer-to-peer approach, in which each participant is a node (Martin, 1998; Reiter and Rubin, 1998;1999). Messages are not wrapped in encrypted layers, but encrypted once.
- lpwa.com, the Lucent personal web assistant (a.k.a. ProxyMate), offered pseudonym services, via which users can easily obtain user names, passwords, and email addresses to be used to access website which require user registration (Gabber, Gibbons, Matias and Mayer, 1997). Lucent has sold this technology to NaviPath.
- www.zeroknowledge.com: Zero Knowledge Systems is an example of a company offering software to enhance privacy (i.e., provide forms of anonymity and pseudonymity).

Note that centralistic services have a greater risk associated with them as they may be (legally) forced to disclose information about their users. Decentralised services do not have this drawback. For humans, two of the most commonly used anonymity services are internet cafés and throw-away web-based email addresses. The first is commonly used to surf anonymously, the latter as a pseudonym.

It is to be expected that services facilitating anonymity and pseudonymity (for agent identities)

- may involve anonymous or pseudonymous agent platforms,
- periodically change the agent identifier of an agent,
- create new agents (possibly on the basis of an example agent) with new identifiers,
- provide an agent platform which effectively hides the agents hosted,
- provide 'obfuscating' agents which employ anonymisation and pseudonymisation services, and
- employ a new technique not yet envisioned.

The chemical commodity marketplace in our example may only allow anonymous agents which are certified by a trusted third party (e.g., to the extent that the agents are able to buy or sell goods), or not allow anonymous agents at all. The marketplace does allow anonymity of users of agents: whether the user of an agent is disclosed during a transaction is up to the agents and the protocols used. The marketplace itself is not anonymous: its needs to have a reputation with its clients to attract clientèle and make profit.

3.5 DISCUSSION

There are two main issues. In the first place, users of software agents have in certain circumstances to comply with identification duties. This seems not to be a difficult problem. Some questions may rise with respect to the modalities, but in general it is something that will sort itself. Secondly, users of software agents may want to be anonymous or only want to be known under a pseudonym. Both legally and technically, anonymity and

pseudonymity cannot be completely guaranteed. Legally, they have not the status of a right (a person does not have a right to be anonymous). Technically, it cannot be guaranteed that a person acting anonymously or pseudonymously.

If the information carried by an agent is confidential it needs to be protected in an open system. A number of techniques for protecting confidential information have been introduced. Anonymity hinges on the principle that information is kept confidential: if at some moment in time an entity discloses the human user of an agent this can be combined with traces of the agent and disclose privacy sensitive information. In addition, anonymity is usually used temporarily, often with a good reason, e.g., when concluding negotiations to finalise a transaction after anonymously gathering information. The consequence of disclosing an identity depends not only on current conversation partners, but also indirectly on the availability of tracing data. Note also that results in mathematics and computer science will most likely make it possible to decrypt currently encrypted information in the future.

Although a number of techniques for protection of confidential information exist, more research is needed about the applicability (when is it most useful), robustness and reliability (when will it fail), and temporality (when will it be deciphered)?

Issues for further research:

- analysis of legal and technical identities of humans and agents
- Must anonymity and/or pseudonymity get a specific legal status? If so, how?
- traceability: analysis of large amounts of tracing data from distributed sources, e.g., in co-operation with analysis of DNS tracing data (http://www.nlnetlabs.nl/dns-analyzer),
- develop realistic, secure environments for agents, including experiments to assess reliability, security, scalability, and feasibility.
- develop identity management models.

4 INTEGRITY & ORIGINALITY

This chapter discusses legal and technical aspects of the recipe for integrity in Section 4.2, integrity of evidentiary data in Section 4.3, and originality in Section 4.4. Section 4.2 on integrity has appeared in a shorter form in the paper "Are Mobile Agents Outlawed Processes?" in LEA-2003 (Brazier, Oskamp, Schellekens and Wijngaards, 2003b).

4.1 ILLUSTRATING SCENARIO

Below, each of the recipes is illustrated by a scenario.

4.1.1 Integrity

In the hospital scenario, software agents are entrusted with the task of archiving patient dossiers and of course making them available to medical personnel when needed. It is clear that the integrity of the information must be guaranteed. It has happened that hackers broke into a hospital's computer system and changed patient data.

4.1.2 Integrity of Evidentiary Data

In the grocery shopping scenario, the baskets in a grocery shop have been fitted with a software agent and a sensor that detects what goods are placed in the basket. The information registered by this software agent is the basis for calculating the amount the user of the basket has to pay when passing the checker. The buyer may not tamper with the data stored by the agent: these data are relied upon for determining the obligations the buyer has towards the grocery shop. Manipulation of data with an evidentiary function is more severely punishable than manipulation of other types of data. Should the producer of the grocer's software agent put in an extra effort to prevent such manipulation?

4.1.3 Originality

Our software agent buys chemicals at a chemical marketplace. The chemicals bought are – at the time of buying – being transported by ship from Asia to Europe. How can our software agent be sure that the seller has not sold the cargo more than once? In the traditional world the seller hands over the original (and therefore unique) bill of lading. But in an on line environment 'originality' and 'uniqueness' do not exist. So how to trade on-line in cargos at sea, such that the buyer knows for sure that he is not being deceived?

4.2 INTEGRITY

This section investigates whether the law protects the integrity of platforms and running agents' processes, and if so how. Technically, it is investigated what measures can be taken to protect the integrity of both the agent and the platform and which counter measures are possible. Questions of law are dealt with according to Dutch law. The words 'computer system' and 'agent platform' are used interchangeably in this section.

4.2.1 Legal Analysis

The proper functioning of software agents is highly dependent upon the conservation of their integrity and the integrity of the platforms on which they function. Integrity means here that no data are unduly altered, erased or supplemented and that the physical objects involved (such as computer systems are not damaged or destroyed).

Data must be taken in a wide sense: they may be data in the strict sense but also code such as software agents or agent platforms. The reason the law occupies itself with the protection of integrity is twofold. On the one hand, the law wants to protect the persons relying on the availability and integrity of data. Their dependence on these data is the rationale for the legal rules in question. This aspect of integrity is addressed in this recipe. A second aspect of integrity concerns the protection of the author of the data. An author can have an interest in maintaining data as he 'created' them. Any subsequent alteration by a third person may affect him in his interests as an author: his moral rights are infringed upon. This aspect of integrity has not been considered.

Integrity raises with respect to mobile agents a number of interesting questions, such as:

- Must a system administrator on whose system a mobile agent runs preserve the integrity of the agent?
- What protection does the law give against attacks by third parties? As an example one may think of a hacker altering patient data that a software agent contains.

For the legal protection of integrity, criminal law is the primary source. For the treatment of the subject, the Convention on Cybercrime of the European Council (hereinafter: CoC) is taken as a starting point. Although this convention addresses the signing states and not the perpetrators of crimes, the convention can nevertheless be used for the purpose of this section, since its provisions are sufficiently clear and precise for understanding what the implementing national provisions will be about.

This section discusses three aspects of integrity: (1) data, (2) bearer of the data, and (3) content.

4.2.1.1 Criminal law protection of integrity

Protection of the data themselves

Article 4 Convention of Cybercrime (CoC) deals with data interference; its text reads as follows:

> Article 4 –Data interference
>
> 1 Each Party shall adopt such legislative and other measures as may be necessary to establish as criminal offences under its domestic law, when committed intentionally, the damaging, deletion, deterioration, alteration or suppression of computer data without right.
>
> 2 A Party may reserve the right to require that the conduct described in paragraph 1 result in serious harm.

In the Netherlands this provision probably requires no implementation action, since Articles 350a and 350b of the Dutch Criminal Code (hereinafter: DCC) already provide for a sufficient criminalisation of the behaviour covered by art. 4 CoC. The Dutch criminalisation even takes a further step than required by the Convention since not only intentional data interference is an offence, but also data interference caused by negligence. The provision about negligence was added, because of the enormous financial consequences that can be the result of negligent behaviour. [Kamerstukken II 1991/92, 21551, nr. 17 Amendement and Handelingen II 11991/92, p. 94-5992] In the provision about negligent interference a requirement of serious harm is included, (compare the option in the second subsection of Article 4 CoC).

What implications does this provision have, for example, for sysadmins who host other peoples mobile agents? Are they allowed to change the content of a hosted mobile agent? This question will be dealt with according to Dutch law. The main question that has to be answered is when and whether the sysadmin acts without right. In the first place, it has to be investigated whether the sysadmin can derive a 'right' from the fact that he owns the system on which the agent runs. After all, in general, a party would be allowed to change the data that are on a computer system he owns. In this respect, the judgement of Hof Amsterdam 18 July 2002 LJN- nummer AE5514 (ABFAB/Xs4all) is relevant. This case centres around the question whether an ISP (Xs4all) can require a sender of commercial e-mail to refrain from sending mail to subscribers of the ISP. Xs4all based her contention that an ISP may indeed require so on the grounds of her ownership of computer capacity, transmission capacity and subscriber collection. The court held, however, that the nature of the service that Xs4all offered implied that the public must have the possibility to send e-mail to her subscribers. This and the fact that e-mail is of increasing importance in society made that the sending of e-mail cannot easily be said to be a violation of Xs4all's right in the computer capacity, etc. It fits in better with the public character of the service-provision to accept that the freedom that Xs4all claims for herself as an owner or titleholder is bound to certain restrictions.

For a sysadmin (modifying an agent) this means that a reference to his (employer's) ownership of the system may not be enough to establish that he did not act 'without right'. This is in particular the case if the service is public in nature and is relevant in society. The sysadmin will have to bring in additional circumstances in order to construe his authority to touch the integrity of the mobile agent. Such additional circumstances may be found in the reason the sysadmin has to modify the agent, e.g., to maintain system integrity. On the other hand, other circumstances may be brought by that weaken his position. A recognisable interest of the user of the agent in maintaining its integrity may, e.g., diminish the authority to affect the integrity of the agent.

However, a remark is in place. The Xs4all/ABFAB case shows a marked difference with the case of the mobile agent. Xs4ALL sought to forbid ABFAB to send mail to XS4all subscribers; thus, Xs4all tried to gain influence on the behaviour of a third party. This is not the same as merely removing the mail stemming from ABFAB from the server or modifying an agent that is present on the system; Xs4all tried to accomplish more. This may have its bearing on the circumstances that have to be adduced to establish authority to modify and may give the

sysadmin more leeway in dealing with software agents. From this observation, it also follows what the sysadmin exactly does with the software agent is relevant. Does he merely modify the agent in a way that is not relevant to the user? Is the user warned about the modification? Does the sysadmin remove the agent-process? If so, does he store all data that enable the agent to proceed on another system casu quo, does he give the agent the opportunity to store these data?

In short, the following relevant factors can be distinguished:

- The nature of the service of the sysadmin (Public or not? Relevant for society?)
- The recognisable interests of both parties (e.g., system integrity and (correct) functioning of the agent)
- The nature and seriousness of the intervention (e.g., modification or removal? Warning the user or not? Allowing the software agent to store its state or not?)

4.2.1.2 The bearer of the data

Above Article 4 CoC was shown to directly protect data against infringement of its integrity. It is not always clear under what circumstances an external party has the right to 'touch' the data. Such problems are much less prevalent in the provisions that are dealt with respect to agents. Simply put, the provisions criminalise penetration of the container in which the data are stored, transported or processed. Two articles of the Convention on Cybercrime are relevant:

> Article 2 – Illegal access
>
> Each Party shall adopt such legislative and other measures as may be necessary to establish as criminal offences under its domestic law, when committed intentionally, the access to the whole or any part of a computer system without right. A Party may require that the offence be committed by infringing security measures, with the intent of obtaining computer data or other dishonest intent, or in relation to a computer system that is connected to another computer system.
>
> Article 3 – Illegal interception
>
> Each Party shall adopt such legislative and other measures as may be necessary to establish as criminal offences under its domestic law, when committed intentionally, the interception without right, made by technical means, of non-public transmissions of computer data to, from or within a computer system, including electromagnetic emissions from a computer system carrying such computer data. A Party may require that the offence be committed with dishonest intent, or in relation to a computer system that is connected to another computer system.

Again both articles do not seem to require implementing action by the Dutch legislator.

The articles do probably not provide protection against a sysadmin that infringes on the integrity of the agent running on his system. The articles only address the access that a sysadmin seeks to the software agent, preceding a possible affection of its integrity. In accessing his system (or the part of his system where the agent is), the sysadmin does not act without right (see Article 2 CoC). The two articles will therefore mainly have a function in protection against hackers that access a computer or intercept a data transmission with the help of a technical means.

The CoC leaves it up to the states whether they want to require that security measures have been infringed. Currently, the Dutch Criminal Code requires that security measures have been infringed. However, this requirement is only one of two options. If the other requirement has been fulfilled, the requirement that a security measure has been infringed need not be met. The other option is: access has been gained by technical interference, with the help of forged signals or a forged key or by assuming a false identity.

Also in the European Union a security measure need not be infringed if an alternative requirement has been met, has found acclaim. See, e.g., the Proposal for a Council Framework Decision on attacks against information systems [COM/2002/0173 final - CNS 2002/0086, Official Journal C 203 E , 27/08/2002 P. 0109 – 0113] . The European Commission states it as follows:

> 'The Commission does not wish to undermine in any way the importance it attached to the use of effective technical measures to protect information systems. Nevertheless, it is an unfortunate fact that a high proportion of users leave themselves exposed to attacks by not having adequate (or even any) technical protection. To deter attacks against these users, it is necessary that criminal law covers unauthorised access to their systems even though there may not be adequate technical protection for their systems. For this reason, and provided that there is either an intent to cause damage or an intent to result in an economic benefit, there is no requirement that security measures must have been overcome for the offence to have been committed.'

The intention of economic benefit may bring sysadmins closer to criminal liability, but not close enough. Being the sysadmin, they will generally not act without right when entering the part of the system where a software agent is. As discussed above, legal access to this part of the system precludes that sysadmins are found guilty under the hacking provision.

The means for committing data interference, illegal access and interception

Protection against infractions to integrity of data may also be afforded by criminalising activities relating to the means that make such infractions easier. Article 6 CoC is a case in point:

> Article 6 –Misuse of devices
>
> 1 Each Party shall adopt such legislative and other measures as may be necessary to establish as criminal offences under its domestic law, when committed intentionally and without right:
>
> a the production, sale, procurement for use, import, distribution or otherwise making available of:
>
> i a device, including a computer program, designed or adapted primarily for the purpose of committing any of the offences established in accordance with the above Articles 2 through 5;
>
> ii a computer password, access code, or similar data by which the whole or any part of a computer system is capable of being accessed, with intent that it be used for the purpose of committing any of the offences established in Articles 2 through 5; and
>
> b the possession of an item referred to in paragraphs a.i or ii above, with intent that it be used for the purpose of committing any of the offences established in Articles 2 through 5. A Party may require by law that a number of such items be possessed before criminal liability attaches.
>
> 2 This article shall not be interpreted as imposing criminal liability where the production, sale, procurement for use, import, distribution or otherwise making available or possession referred to in paragraph 1 of this article is not for the purpose of committing an offence established in accordance with Articles 2 through 5 of this Convention, such as for the authorised testing or protection of a computer system.
>
> 3 Each Party may reserve the right not to apply paragraph 1 of this article, provided that the reservation does not concern the sale, distribution or otherwise making available of the items referred to in paragraph 1 a.ii of this article.

As of September 2002, no country has made a reservation as meant in paragraph 3.

This provision does not have an equivalent in Dutch law. In the Netherlands, merely complicity to any of the offences in the Articles 2 through 4 is criminalised. The protection afforded by the complicity provision is, however, much less than that that follows from Article 6 CoC (at least if it is implemented). Complicity is after all only punishable if the 'supported' crime is actually committed. This hampers the enforceability. Under the provision in Article 6 CoC, there is a crime even if the supported crime does not materialise.

Furthermore, the DCC criminalises the making available or dissemination of computer viruses. This also covers part of the field of Article 6 CoC. E.g., the dissemination of a computer virus that 'attacks' mobile agents is covered.

The current wording of the Dutch provisions on the dissemination still show a shortcoming that makes that they are only applicable to a specific type of virus: worms. There is, however, a legislative proposal pending that amends this legal lacuna [Kamerstukken II 1998/99, 26671, Computercriminaliteit II].

4.2.1.3 Privacy and telecommunications law

If personal data are involved, the security duties of Directive 95/46/EC and with respect to the telecommunications sector, the security duties of Directive 97/66/EC are relevant. The latter directive has been replaced by Directive 2002/58/EC. [Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications), Official Journal L 201, 31/07/2002 P. 0037 - 0047]

On the basis of Article 16 Directive 95/46/EC a security duty is imposed on the controller (at least the implementing national law does; see Art. 13 and 14 WBP) . The controller is the person who determines the purposes and means of the processing of personal data. The controller must implement appropriate technical and organisational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorised disclosure or access, in particular where the processing involves the transmission of data over a network, and against all other unlawful forms of processing. Having regard to the state of the art and the cost of their implementation, such measures shall ensure a level of security appropriate to the risks represented by the processing and the nature of the data to be protected. The Directive gives some additional rules for the case in which somebody processes the personal data on behalf of the controller.

Directive 97/66/EC provides a likewise rule for providers in the telecommunication sector: the provider of a publicly available telecommunications service must take appropriate technical and organisational measures to safeguard security of its services, if necessary in conjunction with the provider of the public telecommunications network with respect to network security. Having regard to the state of the art and the cost of their implementation, these measures shall ensure a level of security appropriate to the risk presented (Article 4; compare Article 11.3 Tw). A peculiarity for the telecommunications regulation consists in the fact that subscribers must sometimes be informed of extraordinary security related events: in case of a particular risk of a breach of the security of the network, the provider of a publicly available telecommunications service must inform the subscribers concerning such risk and any possible remedies, including the costs involved.

The Dutch government has sent a bill to the Dutch Parliament implementing Directive 2002/58/EC.[Kamerstukken II 2002/03, 28851, nrs.1-2, Wijziging van de Telecommunicatiewet en enkele andere wetten in verband met de implementatie van een nieuw Europees geharmoniseerd regelgevingskader voor elektronische communicatienetwerken en -diensten en de nieuwe dienstenrichtlijn van de Commissie van de Europese Gemeenschappen]. The new implementation bill limits the obligation of the providers of telecommunications services to inform subscribers about the possible means with which the risks (of a breach of security) can be counteracted. They only need to give information to the extent that it concerns measures that the providers is not obliged to take himself.

4.2.1.4 Legal requirements

The parties involved are:

- the user of the software agent
- the sysadmin on whose system the agent operates
- the provider of the agent
- the producer of the agent

Precautionary measures to be taken by the user of the software agent are listed below.

The user must make sure that the used software agents have the following features:

- It is advisable that a software agent can communicate the system requirements for its correct functioning to the system to which it wishes to migrate.
- If an agent is to process personal data, the environment in which the agent functions must not pose a danger to the integrity, completeness and exclusivity of the data. Where protocols exist for determining the safety of agent platforms, the user must make sure his software agent is able to handle such protocols.

The precautionary measures to be taken by the sysadmin on whose system an agent operates casu quo the producer of the agent platform:

- It is advisable that an agent (or its user) is warned about integrity risks of running an agent on the system or perhaps better, an agent that may not function correctly, is disallowed access to the system. If the sysadmin is a telecommunications provider, he has a duty to warn.
- Sysadmin wanting to modify or remove an agent from his system must choose the least burdening intervention possible, e.g., no removal, if an insignificant modification to the agent is sufficient.

The precautionary measures to be taken by the provider of an agent:

- An agent must be delivered in configuration that does not present a risk for the system integrity of the system it may run on.

The precautionary measures to be taken by the producer of an agent:

- An agent must be able to communicate its system requirements.

4.2.2 Technical Perspective

In computer systems, integrity is an important feature: improper alteration of a system (and its assets) should be detectable and recoverable (Anderson, 2001; Tanenbaum and Steen, 2002). The integrity of data, an agent and an agent platform is important in the context of this paper. Data, an agent and an agent platform integrity need to be protected from other agents and agent platforms. The integrity of users and system administrators is not taken into account in this document. The notion of trust is related to integrity and is discussed in Chapter 5.

4.2.2.1 Data integrity

Procedures and mechanisms are needed to preserve integrity of data. Such procedures may involve integrity verification procedures (Anderson, 2001), which extensively use traceability information. A number of situations require procedures, mechanisms, and traceability:

- Data in an agent: the integrity of (possibly confidential) information carried in an agent needs to be protected from other agents, agent platforms, and services. Techniques are described in the next Section 4.2.2.2, e.g., involving encryption of information and audit trails of data modification (e.g., in case of agent containers).
- Communicated data: the integrity of information exchanged between agents (and between an agent and an agent platform, and among agent platforms) needs to be protected. A number of methods exist for reliable, secure, message passing (Tanenbaum and Steen, 2002), usually involving some cryptographic techniques such as digital signatures and PKI.
- Data in an agent platform: the integrity of information in an agent platform needs to be protected from local (and remote) agents, other agent platforms, and services. Similar techniques can be employed as used in agents; see Section 4.2.2.3 below.

Note that integrity of data heavily depends on, and influences, reliability of agents and agent platforms (see Chapter 5).

In the hospital scenario, integrity of data is of paramount importance: e.g., undesired changes to patient dossiers may have fatal consequences. Although techniques may be expensive (in terms of development cost and resource consumption), their application is absolutely necessary.

4.2.2.2 Agent integrity

An agent needs to be protected from unwanted alterations, caused by system malfunctions or malicious entities, e.g., other agents or agent platforms. An altered agent may act differently (potentially maliciously), thereby risking its usefulness and potentially also the reputation of the agent itself and its users. Altering agents which carry confidential information (e.g., their human counterpart's credit card information) may cause more damage than altering an agent which doesn't. Note that some agents are more crucial to the correct functioning of an application than others, when, e.g., an agent in a swarm application (e.g., Holland, 1995; Bonabeau and Theraulaz, 2000) becomes corrupted, the effect may be negligible.

Techniques to detect improper modifications are mainly based on tracing alterations to an agent. A mobile agent may need to carry these traces during its life span if agent platforms 'en route' cannot be trusted. One technique is to use an agent container (Karnik and Tripathi, 2001; Noordende, Brazier and Tanenbaum, 2002) as the unit for transportation of an agent and its associated information, as alterations are securely logged. Alternatively, an agent may regularly visit a trusted third party for an integrity check. To protect the integrity of an agent it is necessary that an agent has a unique identity, which distinguishes an agent from all other agents.

Techniques to recover from improper alterations are usually based on restoring information from a copy (a 'backup') at a secure location. Agents with built-in *fault tolerance* may recover more easily, even if one process of the agent becomes corrupted or is terminated (e.g., Marin, Sens, Briot and Guessoum, 2001). Error correction techniques (e.g., as used for data communication over a telephone line) are rarely used as their effectiveness is limited. If no backup is available, then perhaps information can be restored by replaying past behaviour (including consulting relevant agents and agent platforms).

An agent also needs protection from an agent platform. Unfortunately, an agent platform has enormous power over an agent which cannot easily be restrained. Some solutions advocate special purpose hardware, others develop cryptographic techniques (e.g., Sander and Tschudin, 1998). Protocols requiring agent platforms to provide a rationale for their modifications to agents may also be needed. More research is clearly required.

An agent may need to provide 'a reason' for its actions, e.g., to determine liability of users. For this purpose, an agent may be obliged to keep a trace of information (including orders) received from users and possibly from other agents, together with its responses.

In sum, standards are required for protocols and auditing logs concerning agent integrity; this area requires more research, especially with respect to heterogeneous large-scale open systems.

The hospital scenario is an example in which the integrity of agents needs to be guaranteed: agents of doctors may not fail (or have corrupted data). If failure cannot be prevented, the human users should be notified immediately, as they are need to employ alternative courses of action. The integrity of insurance companies' agents is of a lesser concern.

4.2.2.3 Agent platform integrity

An agent platform hosts agents, possibly from different users and involved in different applications. Agents residing at an agent platform compete for its resources (such as processor cycles, bandwidth, disk space) and services (such as (reliable and secure) communication, mobility, automated updates, streaming connections). Breaches of the integrity of an agent platform may result in malfunctioning of agents hosted, possible to loss of reputation of an agent holder, and possible loss of reputation of an agent platform's holder. Note that an agent platform needs not only be protected from agents it hosts, but also from remote agents and other agent platforms. To preserve integrity of an agent platform, an agent platform needs to have policies for access control, prevention, monitoring and corrective action; usually provided by a system administrator. Existing agent platforms have identified this problem (including FIPA-OS, e.g., Poslad and Calisti, 2000).

In the hospital scenario, a closed system, access of agents to the system is tightly controlled: only specific executable code can be used by software agents, of which the users are clearly identified. In such a closed system, extensive monitoring mechanisms need to be in place to trace changes to important data. Prevention of misuse is easier to achieve, and corrective actions may not only involve agents, but also their users: e.g., agents from an insurance company may repeatedly collect too much data which violates privacy concerns of patients, warranting a legal action against the involved insurance company.

Access control

Policies for access control, i.e., deciding whether an agent is to be hosted by an agent platform, are usually based on an agent's identity, and possibly on information of its user, or other characteristics. In addition, the decision may be influenced by information on reputation or gossip, and observations of the (expected) load of an agent platform itself.

In case of updating the Bidder's Edge database software robots were used to search, amongst others, eBay's database. eBays's system administrator could identify agents such as the Bidder's Edge agents. Their agents were, for example, identified by the IP-addresses from which they originated.

When deciding to host an agent, constraints may be imposed, including permissions (read/write to specific information, execution of other programs) and resource consumption constraints (e.g., Jansen, 2001). In case of eBay for example, it was technically possible for Bidder's Edge agents to extensively access eBay.

Prevention

An agent platform may assign different levels of 'trust' to agents it hosts. To limit possible damage, preventive measures may be taken. One measure is to check an agent's code for viruses and other malignant behaviour; this is, alas, impossible, as formal verification does not provide a fast answer if any, and software which scans for, e.g., viruses can only detect known or similar problems. A more reliable approach is to have an agent only use code which is known to be safe, e.g., by downloading an agent's original code from a trusted source. Another measure is to place the agent in a safer, restricted, execution environment, known as a 'sandbox', e.g., the Java Virtual Machine, with which unwanted code commands can be intercepted. In the case of Ebay, preventive measures were not of relevance, as far as can be deduced from the judgement. Fault tolerance in the specific context of an agent platform needs more research, currently this issue is not explicitly addressed.

Monitoring

To provide a reliable environment for each of its agents, an agent platform needs to monitor itself to detect unwanted behaviour. Often, unwanted behaviour is caused by unexpected resource consumption of correctly functioning agents: e.g., when all agents access specific databases in the system congestion and overloading may occur. Detection of unwanted behaviour is complicated by the level of detail which may be needed, and the overhead of information transfer and processing generated by the monitoring system: a policy is required. It is, in

general, not possible to monitor all actions of all agents in minute detail; but it is possible to monitor resource consumption on computers hosting these agents and use this information for management purposes (e.g., Kephart & Chess, 2003; Mobach, Overeinder, Wijngaards and Brazier, 2003). For example, an agent which increases its use of the processor and sends thousands of messages per time unit, may be involved in an unwanted activity. Monitoring message content may only be feasible in specific cases, and has an impact on the privacy of agents and their users.

Another source of information is from outside entities such as system administrators, agents, trusted third parties, and other agent platforms. This information needs to be assigned a level of trust (see Chapter 5), before a subsequent action is taken.

Corrective actions

An agent platform requires guidelines for which automated corrective action to take in which situation, as too many potential situations may emerge to involve a human system administrator. A misbehaving system administrator's agent may be treated differently than someone else's misbehaving untrusted agent. Possible corrective actions include:

- slowing down an agent by allocating fewer processor-cycles,
- adjusting an agent's permissions and resource allocations, e.g., to limit message sending to once per large-time-unit,
- moving an agent into a sandbox,
- forcing an agent to migrate to another agent platform (possibly with the agent's latest internal state, otherwise causing the agent to revert to its last saved state),
- returning an agent, by stopping the agent and sending the agent (possibly together with last saved state) to human counterpart(s) or its previous agent platform,
- killing an agent (including deleting its identity from location services),
- updating and distributing reputation and trust information with regard to an agent,
- consulting a system administrator.

In a case such as eBay, an agent platform could consider isolating the suspicious Bidder's Edge agents in such a way that is not harmful to eBay's system, possibly while storing enough data in order to allow the agents to resume on another system (this was not an option for eBay). A more drastic option which eBay's system administrator could consider is killing Bidder's Edge agents. In all cases the system administrator could consider notifying the owner of the agents (Bidder's Edge) of their illegal presence and their potential termination.

The agent platform is required to maintain a trace of its corrective actions possibly with a rationale, and may require a protocol for notifying agents or their users related to the subjects of corrective actions. A rationale should specify the (potential) damages to be caused by the agents involved and the appropriateness of the corrective actions taken.

4.3 INTEGRITY OF EVIDENTIARY DATA

The legal part of this recipe deals with the legal rules relating to evidence. The legal analysis consists of three parts: 1: the law of evidence; 2: legal requirements that promote the integrity of evidentiary data; and 3: an evaluation of the legal requirements in light of the use of agents in an electronic environment.

In this legal analysis Dutch law prevails, European regulatory initiatives are, however, included. The relevant Dutch act is the Civil Procedure Act (CPA). The main problem addressed in this part of the recipe is whether electronic data can be entered as evidence into a legal procedure and how this electronic data should be appreciated by the courts.

4.3.1 Legal Analysis

This recipe deals with evidence. From a legal point of view, one can ascertain whether electronic data is admissible in a legal procedure. In general, the question can be raised what evidence is allowed by law. In theory, two different approaches can be distinguished. Firstly, one can imagine a regulation of evidence which allows for all kinds of evidence to be allowed in a civil procedure. The *appreciation* of this evidence however, is at the discretion of the court. Secondly, a different approach might be that a statute prescribes in detail under which circumstances evidence is permitted *and* how this evidence must be appreciated by the court.

4.3.1.1 Law of evidence

The starting point for the Dutch law of evidence is that evidence can be supplied by any means, unless a statute provides otherwise, Article 152 subsection 1 CPA. Therefore, the use of electronic data as evidence should not pose a problem. However, the *appreciation* of the evidence is solely at the discretion of the judge (unless a statute provides otherwise), Article 153 CPA. There are instances, however, in which courts must consider certain types of evidence as a reflection of the truth. This can be called *coercive evidence*. Evidence to the contrary however, is admissible, unless a statute provides otherwise.

In short, Dutch Law starts from the premise that evidence can be furnished by all means. In addition to this, the CPA lays down rules for specific evidentiary means (Kemna, 2001). The following means are dealt with.

1. Instruments;
2. Witnesses;
3. Recognition;
4. Declarations by experts;
5. Court inspection on the spot.

With regard to the first means, the so-called *instruments* should be explained. An instrument is a *signed writing*, intended to serve as evidence, Article 156 subsection 1 CPA. Therefore, as to the physical form of the document, an instrument consists of a *writing* and one or more *signature(s)*. According to subsection 2 of the same article, an *authentic* instrument is a instrument which is drawn up in the required form by an authorised notary. On the other hand, instruments executed privately are all instruments that are not authentic instruments. According to Article 157 CPA, authentic instruments furnish coercive evidence against everyone of that which the notary has declared, within his authority, with regard to his observations and actions. Furthermore, authentic instruments, as well as instruments executed privately, furnish coercive evidence with regard to a declaration by a party concerning that which the instrument is meant to prove.

Instruments have three kinds of evidentiary value (Hidma and Rutgers, 1995):

1. Outward evidentiary value (or: *presumption of authenticity* (Dijksterhuis-Wieten, 1998; Hidma and Rutgers, 1995): a writing that appears to be a authentic instrument will be considered to be an instrument until otherwise is proven;
2. Formal evidentiary value: it is assumed that it is true that a declaration was uttered as it is written in the instrument;
3. Substantive evidentiary value: it is assumed that *what* was declared is true.

4.3.1.2 Legal requirements promoting the production and integrity of evidentiary data

In Dutch law there are a number of statutory rules that demand that in certain instances evidence must be produced and/or retained. The following two regulations can be mentioned.

1. Rules regarding the issuing of receipts;
2. Rules regarding the keeping of an adequate administration.

Issuing receipts

There are two regulations because of which a receipt must be issued. Firstly, there is the regulation based on the Dutch Civil Code (DCC). According to Article 6:48 DCC there is a general duty of the creditor to issue a quittance for each payment, unless contract, usage or equity produce a different result. Furthermore, if the creditor has documentary evidence of the debt, the debtor can, upon payment, also claim that this documentary evidence be given to him. However, the creditor can keep the evidence, if he has a reasonable interest in keeping it and inscribes the necessary annotation on it as proof of the release of the debtor. If the creditor refuses to issue a quittance, the debtor can suspend the performance of his obligation.

Secondly, according to VAT (Value Added Tax) regulations, the entrepreneur has to supply his customer with a numbered and dated receipt. Apparently, this must be a paper (non-electronic) receipt. In certain situations however, entrepreneurs are allowed to issue the receipt using an electronic message (Besluit van 16 oktober 1997, nr. 423 DGM, Vakstudie Nieuws 1998/19.21), if (amongst others) the following requirements are met:

- the receipt-data that is to be exchanged between computers must be structured according to certain agreements or norms. This requirement will be met in case of EDI.
- the electronic message must contain, amongst others, the following information:
  - the day of performance of the service or delivery of the good;
  - name and address of the entrepreneur performing the service or delivering the good;
  - name and address of the customer;
  - a clear description of the good or service;

- the quantity of the goods;
- in certain instances the VAT identification number of the entrepreneur and that of the customer;
- the price of the good or service;
- the amount of VAT that is owed.

- both parties (the entrepreneur and the customer) need to apply for a licence.

As to the retention of these electronically delivered receipts, the following general regulation with regard to the keeping of a adequate administration applies.

Keeping an adequate administration

Most legal systems impose a general duty to keep a administration upon businesses and professionals. The DCC states in the Articles 3:15a and 2:10 that anyone practising a profession or trade, must keep an administration of his financial position. All documents, records and data carriers must be kept in such a way that the rights and obligations are clear at all time. These documents, records and data carriers must be retained for a period of seven years. The data contained on a data carrier can be transferred to and kept on another data carrier, on the condition that the transferral occurs with correct and complete rendering of the data and this data during the retaining period remains completely available and can be made readable within a reasonable amount of time.

4.3.1.3 Evaluation

As discussed above, the law requires in certain instances that evidence be collected, stored and made available. Can this be done on line with the help of software agents? Not all forms of evidence that are available off-line have an on-line equivalent. The current legal situation does, e.g., not allow for an electronic instrument.[5] This is, however, not such a big problem as it may seem. According to Dutch law, evidence can namely be furnished using all lawfully available information, be it that a court is free in its assessment of such evidence. This means that a party delivering electronic evidence must convince a court of its value. In order to make the evidence more convincing during a court of law, parties might want to take technical measures to ensure the integrity of the data. This way, the judge will assign greater evidentiary value to the data. Parties might even conclude a contract regarding the evidentiary value of certain data. This also holds for data with evidentiary value. The grocer mentioned in the scenario is not obliged to take extra measures to prevent manipulation of the data inside the shopping cart agent. This does, however, not take away that not taking measures may make it afterwards more difficult to convince a court of the evidentiary value of the information one presents as evidence. Furthermore, every manipulation that has been prevented through good physical or logical security is one fraud less!

The rules regarding the obligation to keep an administration do not appear to be a barrier to the use of electronic agents. Data is allowed to be retained in electronic form, on the condition that certain requirements are met. The requirement to issue VAT-receipts seems to form more of a barrier. Although it is possible to issue these receipts electronically, the requirement of a license that must be applied for first, can undo the advantages that the electronic exchange of data brings.

4.3.1.4 Legal requirements

The following parties are involved:

- The party furnishing electronic evidence
- The party relying on electronic evidence
- The provider of the agent
- The producer of the agent

The measures to be taken by the party furnishing electronic evidence (and using a software agent for this purpose):

- Duties to provide another party with evidence, must be adhered to; examples: the creditor furnishes a quittance upon performance by the debtor and the entrepreneur provides his customer with a numbered and dated receipt.
- In certain cases of a duty to provide another party with evidence, the form and contents of the evidence is prescribed: see the requirements for a VAT-receipt. These requirements must be adhered to.
- Businesses and professionals must adhere to the duty to keep an administration of their financial position.

The measures to be taken by the party relying on electronic evidence (and using a software agent for evidence collection etc.):

[5] This might change in the near future, Kamerstukken I 2002/03, 27 743, nr. 35, p. 10.

- The relying party must ensure that it gets the evidence, it may need at a later point in time, or take appropriate measures, e.g., a debtor must make sure he gets a quittance upon performance of his obligation, or suspend the performance.
- The integrity of the evidence (that is to be presented in court) must be guaranteed as good as is reasonably possible. Reason: if the integrity of the material (e.g., loggings) that is to serve as evidence is not or insufficiently guaranteed, this may have negative consequences when it comes to an assessment of its evidentiary value.
- The process of evidence collection, storage, retrieval and presentation must be transparent and verifiable. Reason: it is no use to have integer evidence, if proving its integrity is impossible.

The precautionary measures to be taken by the provider of the agent:

- The provider of the software agent informs the user about the functioning of the evidence gathering functionality of the software agent, especially if this relevant for their correct functioning. If the evidence data are not stored in the mobile agent but on the computer of the user, the user must e.g., be told that his computer must be on line and have enough memory capacity to store the evidence data.
- The provider must indicate for what purposes the evidence gathering facilities can be used. The quality of the functionality may e.g., not be sufficient for high value transaction. The duty to inform is more ponderous if the provider knows more about the use the user intends to make of the software agent.

The precautionary measures to be taken by the producer of the agent:

- The information duties mentioned with the provider also hold for the producer.
- The processes of evidence gathering, storage, retrieval and presentation must be integer.
- The integrity of said processes must be demonstrable.

4.3.2 Technical Perspective

First technical aspects of evidentiary data are discussed, then technical aspects of integrity and finally the parties involved.

4.3.2.1 Evidentiary data

Evidentiary data is quite similar to tracing data; both commonly include the following information: identifier (issued by writer of the datum), subject (what is it about), circumstances (when, where), and parties (who are involved: agents and possibly their users). The terms evidentiary data and traceability are used interchangeably.

Traceability already plays an important role in Internet applications; e.g., traceability of money systems (Sherif, 2000) is used for merchandise delivery and dispute arbitration. Audit logs, another form of trace data, are used in, e.g., bookkeeping systems, to securely track changes to sensitive data such as banking accounts (Anderson, 2001).

Traces may not need to contain many details, traffic analysis (when is a message sent to whom, irrespective of content) may yield sufficient detail to be used in court. Traceability, however, conflicts with the desires for non-excessiveness, anonymity, identity protection, … Tracing data can be used as evidence in court, but also for blackmailing. This implies that traces need to be confidential, only disclosable to authorised parties. It is important for users to know whom these parties may be and whether they are trusted, e.g., see the issues with security breaches in Microsoft's Passport technology (McWilliams, 2001), which contains people's identificatory information such as creditcards and home addresses. In addition, the effect of coupling traces from a number of sources is unknown (cf. privacy issues when linking databases), as is the durability of traces.

The distributed nature of traces in large-scale agent systems causes problems, including:

- different formats of traced information,
- different amount of detail in traces,
- different legislations,
- different notion of time in traces (no global time on the Internet),
- different protocols for tracing,
- different protocols for removing old traces,
- large amounts of tracing data.

Acquiring evidentiary data is much facilitated in a closed system such as in the grocery shopping scenario: all participants can agree on the format, detail, and protocols to be used. In open systems, e.g., the chemical commodities marketplace, acquisition of evidentiary data is more difficult: a marketplace needs to stipulate its policies and have clients (buyers and sellers) agree (perhaps to the extent that clients provide evidentiary data when requested).

4.3.2.2 Integrity

Maintaining integrity of data involves the same mechanisms described concerning identity protection, as well as a number of additional mechanisms:

- Non-repudiation: what is logged, is 'true'.
- Authenticity: the evidence data is produced by the logging system, and no one else.
- Storage: evidentiary data needs to be securely stored, recoverable in the event of mishap.
- Integrity: evidentiary data needs to be integer, changes need to be detected, e.g., by using digital signatures.
- Confidentiality: evidentiary data needs to be encrypted, being only readable by intended parties.
- Verifiability: evidentiary data needs to be verifiable, to support its credibility and originality.
- Protocols: are needed to describe what is to be logged, how it should be stored, by whom, when, and how the logging can be verified.
- Audit trails: mechanisms are needed which irrevocably add date and time stamps to any entry logged, as well as other information.
- Social protocols: the role of human administrators, who possibly have access to evidentiary data, needs to be described, and logged as well.

4.3.2.3 Role of parties involved

Many parties may generate tracing data, possibly to be used as evidentiary data. These include:

- outsiders, e.g., by analysing communication and action traffic of agent platforms, and potentially of individual agent(s).
- trusted third parties, which may authenticate, authorise, and verify evidentiary data,
- agent platforms, tracing local agent's behaviour in interaction with other agent platforms and remote agents.
- Agents, tracing their own actions and interactions with agent platforms and other agents.

An important issue concerns the information that needs to be included in a trace: what level of detail is needed, when is which information needed, and how long should it be kept? Protocols and standardisation are required:

- The party which furnishes electronic evidence needs protocols outlining when evidence provision duties are in play, and which formats are required as well as protocols concerning tracing their own actions.
- The party which relies on electronic evidence needs protocols for evidence collection and transparent and verifiable storage.
- The party which provides the agent needs protocols to inform users about the (quality of) functioning of the agent with respect to evidence furnishing, collecting, and storing.
- In addition, the party which produces the agent needs protocols and techniques to validate and verify correct functioning of the agent with respect to evidence furnishing, collecting, and storing.

Enforced tracing, in, e.g., agent platforms or in agents, may not always yield reliable data: the writer's of agent software and agent platform software may devise means to circumvent tracing modules. Enforcing a rigorous internal structure on agents or agent platforms to facilitate tracing, may severely hamper development, deployment, usage, and acceptance of agents and agent platforms.

4.4 ORIGINALITY

Legally, originality plays an important role as an assurance of authenticity and unicity. Originality can, however, only exist if copies can be distinguished from originals. This immediately shows the great difficulty encountered when using digital data: copies are identical to the 'originals' that functioned as the templates from which the copies were made. In a digital environment the authenticity and unicity of data must thus be guaranteed in an alternative way.

4.4.1 Legal Analysis

Originality refers to the state of not being a copy. In the traditional 'paper' environment, originality can generally, easily be detected, since it is difficult to create an exact replica of a paper document. Copies made with the use of generally available copying techniques (such as photocopying) often show clear quality differences with the original. With respect to digital data, it is not clear how 'originality' could be defined and even, whether 'originality' should be defined. Perhaps, one could define the original as the first set of digital data that came into being. Later replicas of digital data are, however, not distinguishable from the 'original'. Therefore, this concept of originality seems less suitable for use in practice, unless additional measures are taken

in order to make the 'originality' perceptible. Time stamping could be such a measure; it may, however, not take away all questions.

The main functions of originality in law are twofold:

- Originality has an evidentiary dimension to it. Originality is seen as a token of the authenticity of a document and thus adds to its evidentiary value. The Dutch private instrument and authentic deed enjoy e.g., their special evidentiary status only if they are originals. There are a few exceptions to this rule, but they will not be dealt with here.
- Original documents can be used as documents of title, such as bills of lading. It is the uniqueness of original documents that makes them suitable for such use. A carrier will, e.g., deliver the goods to the person that is able to surrender an original bill of lading. Since a cargo can be delivered only once, it is important that not too many documents circulate that can pass off as original bills of lading.

4.4.1.1 The evidentiary function

Since originality is undefined or little useable with respect to digital documents, the special evidentiary value law binds to original paper documents does not apply to digital documents. The fact that originality is not a means for heightening the evidentiary value of digital documents, does of course not take away that the reliability of such documents can be enlarged by other means, such as the use of (asymmetric) encryption, good logging practices, the use of "WORM" media ("write-once, read-many"), etc. The use of such technologies may convince a judge or other arbitrator of the evidentiary value that can be attached to such means of evidence. For the time being, statutory law does only to a limited extent bind a fixed evidentiary value to the use of such technologies. The Directive on electronic signatures gives, e.g., a special status to advanced electronic signatures, certified by a qualified certification authority. Such a careful approach of the legislator is only wise, given the little experience that exists with respect to technologies that heighten the evidentiary value.

4.4.1.2 The 'uniqueness' function of originality

This second function will here be illustrated with the help of the example of the bill of lading, which may in practice be the most frequently occurring application of an original document as a document of title.

The three main functions of a bill of lading are the following:

- Receipt, the carrier hands the shipper the bill of lading upon receipt of the goods. The bill of lading confirms the receipt, contains a description of the goods and the state in which they are received.
- Contract of carriage, the bill of lading contains the carriage contract
- Negotiable paper; the bill of lading is a negotiable paper allowing for the transfer of the cargo while at sea. At the port of destination, the carrier will deliver the goods to the person who surrenders an original bill of lading to the carrier.

In this section, the focus is on the bill of lading because of its third function, as this is illustrative with respect to originality. For those interested in other aspects of bills of lading, please refer to the traditional literature on the subject, e.g., (Proctor, 1997). As stated, an original bill of lading is an instrument for transferring the possession rights in a cargo to a third party and the presentation of an original bill of lading is the means to obtain delivery of the goods. The carrier may only deliver the goods against the production of an original bill of lading. He may thus not deliver the cargo without any bill of lading being produced, or against a copy. This protects the shipper or subsequent owner of the cargo against insolvency of the buyer; after all, the original bill of lading is only transferred after payment or against provision of sufficient financial security. This is the very point highlighted in the scenario at the beginning of this chapter.

As is apparent from the English case of Motis/Dampskibselskabet, the requirement of the original bill of lading also serves to protect against fraud [See Motis Exports v Dampskibselskabet AF 1912 [1999] 1 Lloyd's Rep. 837]. In this case, a carrier had delivered the goods to the holder of a forged bill of lading. Although the forgery was not apparent, the carrier was held liable for the loss suffered by the holder of the original bill of lading. The decision was appealed against, but the court of appeal upheld the decision of the Commercial court [see Motis Exports v Dampskibselskabet AF 1912 [2000] 1 Lloyd's Rep. 211]:

> 20. In my judgment Mr Justice Rix was correct to characterize what occurred as misdelivery. A forged bill of lading is in the eyes of the law a nullity; it is simply a piece of paper with writing on it, which has no effect whatever. That being so delivery of the goods, or in this case the delivery order which was tantamount to the delivery of the goods, was not in exchange for the original bill of lading but for a worthless piece of paper.

The carrier thus may be very well advised to inspect a surrendered bill of lading in order to determine whether it is genuine or a forgery.

The use of paper bills of lading does, however, carry important disadvantages. It is, e.g., not uncommon that a cargo arrives at the port of destination before the bill of lading does. In order to be able to unload the vessel, the carrier is usually prepared to do so against a letter of indemnity. This is, however, no more than a makeshift measure. This drawback can be counteracted by digitising the functions of the traditional paper bill of lading and the processes of handling it. The digitalization would also open up the possibility to support the processes using software agents.

From a legal perspective, such a digitalization project encounters two main problems.

In the first, place there is insecurity with respect to the applicability of existing legislation about bills of lading or more generally documents of title, since existing law contains traditional terminology, such as 'document', 'writing' and 'signature'. The existing statutory law can be found in international conventions, such as the Hague-Visby Rules and the Hamburg Rules or the 'Verdrag ter vaststelling van enige eenvormige regelen betreffende cognossement; Brussel 25 augustus 1924 (Trb. 1953, 109; Nederlandse vertaling: Trb. 1957, 24) Cognossementsverdrag (plus wijzigingsprotocollen). Furthermore, such law can be found in national statutes, such as The Carriage of Goods by Sea Act of 1992 (in the UK) or Boek 8 of the Dutch Civil Code (hereinafter: DCC). Wording such as signature, writing or document renders their applicability to electronic bills of lading unsure.

In the second place, there can be doubt about the suitability of electronic messages as a means to perform the traditional function of bills of lading, because of the differing physical characteristics of electronic messages as compared to paper documents. The supposed characteristic of paper documents that exists in the fact that originals can be distinguished from copies makes that original paper documents are recognizably unique: a copy is recognized as such (at least that is the commonly used presumption). This uniqueness of original paper documents is, however, only relative: in practice, often more than one original is prepared. That is no problem, as long as the paper documents do not get in the hands of persons that may not have the documents at their disposal. The singular character of a paper document facilitates that the disposition over documents can be held exclusive. Digital data on the other hand are multiple. The transfer of digital data to a receiver does not imply that the sender no longer possesses the data. This characteristic could entail that something like an 'electronic bills of lading' becomes more widely dispersed than is desirable. The relative uniqueness that is part and parcel of 'original' documents is not present with documents or messages that contain digital data.

The doubt about the physical suitability may even reinforce the insecurity with respect to the applicability of statutory provisions, mentioned before. The doubt about the physical suitability may, however, be taken away or lessened by building more security into the processes, used to handle 'electronic bills of lading'. Hereinafter, is shown that Bolero directs all messages through a central messaging platform, thus enabling various kinds of checks.

The problem with respect to the applicability of the statutory provisions may be solved in two alternative or cumulative ways:

1. It is conceivable that existing legislation is adapted. The Workgroup on electronic commerce of UNCITRAL has done much work in this respect.
2. The same functionality may be performed while making use of other legal concepts and constructs. The legal forms may then be chosen in such a way that national and international legislation does not constitute an obstacle. Hereinafter, it is shown that Bolero has brought all processes with respect to her 'Bolero Bill of Lading' (hereinafter: BBL) into a contractual framework, that functions as a layer that isolates the BBL somewhat from national and international legislation.

Ad 1.

The UNCITRAL model law on electronic commerce contains two articles on transport documentation; its scope is thus somewhat wider than bills of lading alone (Clift, 1999). The core provision can be found in article 17 section 3. It reads as follows:

> (3) If a right is to be granted to, or an obligation is to be acquired by, one person and no other person, and if the law requires that, in order to effect this, the right or obligation must be conveyed to that person by the transfer, or use of, a paper document, that requirement is met if the right or obligation is conveyed by using one or more data messages, provided that a reliable method is used to render such data message or messages unique.

Note that the place of a paper document can be taken by one or more data messages. It is, however, not required that one of these messages takes the place of 'the' paper document or that it is 'the' electronic bill of lading. The

messages need also not be unique in themselves, but they may derive their uniqueness from the procedure in which they are used. The model law thus leaves some room to adapt the processes around transport documentation to the characteristic traits of ICT.

The method that renders the messages unique must be reliable. This reliability is to be assessed in the light of the purpose for which the right or obligation was conveyed and in the light of all circumstances, including any relevant agreement.

Ad 2.

The use of other legal concepts and constructs than the 'bill of lading' is illustrated with the help of Bolero.

> Bolero is not the first initiative to take bills of lading to the digital environment (Burnett, 2001; Laryea, 2001; Faber, 1996). Previous initiatives include the CMI Rules and SeaDocs. Bolero is focused on as it seems to be the most viable of initiatives. Other initiatives, such as Trade Card are not discussed.

All parties involved enter into contractual relations with Bolero International Ltd. and Bolero Association Ltd. The latter agreement is accompanied by the Bolero Rule Book and operational procedures. All electronic communications are encrypted using public key encryption. The contractual agreements lay a basis for the legal validity of the communications. Communications do not take place from party to party directly, but involve the title registry, which thus is in a central position allowing it to keep an eye to 'uniqueness'.

As an illustration, a trace of a Bolero Bill of Lading is described here:

When receiving a cargo, the carrier creates a so-called Bolero Bill of Lading (hereinafter: BBL). The BBL is an electronic document containing the same information as a traditional bill of lading. The carrier sends the BBL to the Bolero International's title registry. The title registry checks the digital signature of the carrier, registers the BBL in the registry and passes the BBL on to the shipper. If the shipper or a subsequent holder of the BBL wishes to transfer the cargo while on sea, he passes the BBL on to the transferee and notifies the registry. Upon notification, the title of the current holder is cancelled and the title is transferred to the next holder. The new holder has 24 hours to inform the registry that he accepts that he is the new holder. Legally, a new carriage agreement between the new holder and carrier arises through novation. At the same time, Bolero International informs the new holder on behalf of the carrier that the carrier holds the goods to the new holder's order. Legally, this constitutes attornment.

If the goods arrive at the port of destination, the holder surrenders his BBL to Bolero International. Bolero International notifies the carrier of the surrender and confirms the surrender to the 'holder'. The 'holder' can collect the goods at the port of destination upon identification as required by the carrier or the port.

Every communication involves Bolero International. It checks the digital signatures of the messages it receives and checks the contents of the messages against the title registry. It is through the procedures employed (and not the electronic messages per se) that the uniqueness of the 'holder' is guaranteed. Originality is thus reduced to its core element in the context of negotiable papers: uniqueness. The uniqueness is guaranteed in a way that fits in with the special characteristics of ICT.

4.4.1.3 Legal requirements

Because 'originality' does not exist in an electronic environment, our approach is to use the solution found by Bolero as a starting point with respect to the measures that the parties must take, i.e., the legal function of originality can be fulfilled by participating within the Bolero framework.

The following parties are involved:

- The party entitled to a performance (delivery of a cargo)
- The party obliged to perform (to deliver a cargo)

The measures to be taken by a party entitled to a cargo (participating in Bolero):

- The software agent must be able to provide information to other participants: e.g., the surrender (informing the Title Registry that one wants to take delivery of the cargo)
- The software agent must be able to verify facts: Does the carrier hold the cargo to the holder's order? A message to that effect must be received.
- For the communications mentioned, it is necessary that the software agent is able to reliably authenticate messages with the electronic signature of its user. Furthermore, the software agent must be able to communicate with the Title Registry and other participants.

The measures to be taken by a party obliged to deliver a cargo (participating in Bolero):

- The software agent must be able to provide information to other participants: e.g., delivery of the BBL to the buyer.
- The software agent must be able verify facts: Does the buyer accept the transfer of the cargo? A message to that effect may be received.
- For the communication mentioned, it is necessary that the software agent is able to reliably authenticate its messages with the electronic signature of its user and verify the signature of other participants (mainly the Title Registry). Furthermore, the software agent must be able to communicate with the Title Registry and other participants.

4.4.2 Technical Perspective

Originality of digital information is related to watermarking issues, which is about marking information in such a way that the copyright holder can be identified. The Bolero approach is also about ownership transfer, which means that some digital information (possibly describing a real-world artefact) is to be owned by a different (legal) entity.

The Bolero approach basically describes a transaction between two software agents. Briefly summarised, Bolero (http://bolero.net, e.g., Edwards, 1996) is about secure communication between two parties, transmission of proof of ownership (possibly by using a certificate), and transfer of ownership (possibly by using a certificate and involving a trusted third party). This involves the use of protocols, digital signatures, certificates, verification techniques, traceability and a central certification authority.

Common approaches in electronic transactions, e.g., involving credit cards, employ trusted third parties to 'witness' ownership transfer. It is important in this context that the previous owner abandons ownership. An approach about change of 'ownership' is in the context of agent migration, in which an agent's code and data are transferred to another agent platform in an agent container (Karnik and Tripathi, 2001; Noordende, Brazier and Tanenbaum, 2002). Each agent platform may change this agent container, but needs to digitally sign these changes, resulting in a change log which is signed by agent platforms visited by the agent (hereby ignoring privacy issues), providing proof of what happened when. Such a trail of ownership (of the *n*-previous owners) may be useful in transferring ownership of digital information. Other approaches may involve decentralised authorities, in which peers are involved in 'witnessing' and 'asserting' originality of certificates on behalf of other agents: webs-of-trust, e.g., as used in PGP (Williams, 1995).

It is impossible to prove, in general, that the current owner of digital information is the only owner (and the real owner), if the digital information is not related to the proof-of-ownership, e.g., via a trusted third party. This may require changing the digital information whenever ownership is transferred. Protocols are needed to solve issues such as old copies of digital information by previous owners, including copies of proof of ownership by previous owners. However, solutions may come from research in the area of digital money (e-cash, virtual cash, etc, see Sherif, 2000): different agents may currently 'own' digital money, and transfer this money among themselves.

In the chemical commodities marketplace, the marketplace needs to distinguish between matchmaking (bringing buyers and sellers together) and transactions (transfer of money and goods). The latter requires that our marketplace uses an approach which is supported (and approved) by its clients: e.g., Bolero.

4.5 DISCUSSION

The conditions under which access to a computer system is granted are relevant. Nevertheless, a number of outstanding questions with respect to these conditions remain. The conditions for use of the system must, e.g., be clearly communicated to the software agent and its user by the system administrator. A straightforward way to 'communicate' the conditions is perhaps a system that is able to 'physically' prohibit any use of the system that violates the conditions. If a platform can actively prohibit usage which violates the conditions, then a question is whether a platform can distinguish between usage satisfying or violating the conditions.

Technical measures may have an all or nothing character: allowing either too much or too less. In the eBay v. BE case, the former blocked at one point the latter's IP-addresses in an attempt to prohibit the uses that violated the conditions. This measure did, however, not differentiate between uses that satisfied or violated the conditions. This case also highlights a second drawback of physical measures: BE easily robbed the measure of its effectiveness by using a proxy-server, thus hiding her IP-address and nullifying the effect of eBay's IP-blocking. In addition, technical measures may, on their own, make the precise conditions for use insufficiently explicit: was, e.g., BE's access through the proxy-server allowed since it was not blocked or must IP-blocking be understood to be a general denial of access?

A protocol that facilitates communication of the conditions of use much more clearly and explicitly than is the case with 'coercive technical measures' is needed. A distinction needs to be made with respect to the recipient of the conditions of use. If they are to be communicated to the software agent, a protocol is required that can express that conditions are being communicated and specify these conditions, in such a way that this is meaningful for the software agent. In addition, it is desirable for communication of conditions to guarantee non-repudiation of receipt, i.e., it must be provable that the software agent has undeniably received and understood the conditions. An appropriate protocol could, e.g., require a software agent to acknowledge receipt of the conditions. If the conditions are to be communicated to the owner of the software agent, the owner must be identifiable and have a contact address. Again, a protocol is necessary, including non-repudiation of receipt.

For the time being, system administrators may still be confronted with the presence of 'suspicious' agent's processes on their systems. The technical analysis has shown that there is a whole range of measures a system administrator can take against software agents violating conditions for use. These measures range from reducing resources for the agent to removing the agent from the system. The legal demands of proportionality and subsidiarity imply that the least far-reaching measures that are effective must be chosen. The question is, however, the effects of these measures. The answer to this question may be learned by experience as it largely depends on dependencies between large numbers of co-operating agents and does not lend itself very well for imposition by a legislator. However, learning by experience does not come by of its own. Experiences have to be collected and discussed, a subject that lends itself very well for self-regulation by organisations of system administrators. The outcome of such collection and discussion can be laid down in directives that can guide system administrators in their approach to agent processes that go beyond the latter's allowed use.

5 TRUST

Trust plays a role in both human and automated agents' tasks, including information retrieval, negotiation, co-ordination, transactions, migration, etc. and information (Falcone, Singh and Tan, 2001; Deutsch, 1962; Castelfranchi and Falcone, 1998; Gambetta, 2000). Trust involves the dilemma whether positive results of trusting another agent (or information) outweighs the risks of not trusting another agent (or information).

Commonly, trust of an agent in other agents is based on its own direct and indirect experiences, other agent's experiences, and reputation of an agent (Aberer and Despotovic, 2001; Mui, Mohtashemi and Halberstadt, 2002). Trust is context dependent and dynamic, and is usually not transitive. Trust and distrust are most often reciprocal by nature (Lawson, 1997; Falcone & Castelfranchi, 2001). Trust changes over time as agents continually update beliefs in other agents (Birk 2001; Witkowski, Artikis and Pitt, 2001; Beth, Borcherding and Klein, 1994; Barber and Kim, 2001) and themselves. The more open and dynamic an agent's environment is, the more incentives may differ, and the more agents may provide untruthful information (Jurca and Faltings, 2002, Abdul-Rahman and Hailes, 2000; Schillo, Funk and Rovatsos, 1999; Beth, Borcherding and Klein, 1994).

Trust models are often developed for a specific purpose, and usually express: acquisition and representation of trust, communication of trust, and reasoning about trust. For example, the Bell-LaPadula (Bell and Padula, 1973) model is designed for the military, in which levels of trust are distinguished. Marsh's (1994) trust models, however, distinguish basic, general and situational trust, in which basic trust influences general trust in an agent or event, which is adjusted for a specific situation into situational trust. Castelfranchi and Falcone (2000) and Grandison and Sloman (2000) describe how trust about an agent is related to specific beliefs about the agent, involving beliefs about the agent involving its expertise, reliability, and availability.

An example of a centralised approach for agent trust acquisition and representation is Trustbuilder, a trust management system (Winslett, Yu, Seamons, Hess, Jacobson, Jarvis, Smith and Yu, 2002), in which resource access is regulated, based on both agent's and Trustbuilders policies and credentials. During a negotiation certificates and policies are exchanged until Trustbuilder is satisfied, or the agent is unable to adhere to a policy. A decentralised approach combining a trust model with a peer-to-peer infrastructure (Aberer and Despotovic, 2001) stores complaints about peers, which form the basis for a reputation-based trust scheme. Agents may decide to, e.g., engage in communication if the number of complaints is below a specific threshold. The system is self-regulation to the extent that complaints may also be about truthfulness of other complains.

Trust plays also an important role in an agent platform: trust in agents, trust in other agent platforms, and trust in users. Agent platforms are responsible for maintaining integrity and security, both aspects requiring an assessment of trust (or risk) in agents being hosted. For example, access control (i.e., which agent is allowed to enter under which conditions) may involve deliberations concerning the reputation of an agent, their user, and the producer.

Trust is commonly based on reputation (e.g., based on direct and indirect observations and hearsay), which implies that agents and agent platform use reputations to form trust in others, and may actively pursue actions which increase their own reputation to positively influence the trust others invest in them.

The legal role of trust is difficult to capture. Chapters 2 (Autonomy), 3 (Identifiability and Traceability) and 4 (Integrity and Originality) contain a legal perspective on agents. Adherence to the recipes in Chapters 2, 3 and 4 implicitly and explicitly involves trust: trust in involved parties to 'do the right thing'. Trust is a well-researched topic in agent systems, enabling agents (and agent platforms) to involve trust in their deliberations. When a user did not use agents fitted with trust mechanisms, and something goes awry, the user can be held legally liable. The same holds for system administrators and agent platforms. Stated differently, a user or system administrator can limit his or her liability by employing agents or agent platforms which are fitted with trust mechanisms (involving trust in these trust mechanisms).

In this chapter, three recipes are distinguished with are strongly related to trust: reliability (Section 5.2), confidentiality (Section 5.3), and non-excessiveness (Section 5.4).

5.1 ILLUSTRATING SCENARIO

Each of the recipes is illustrated by a scenario from Appendix A.

5.1.1 Reliability

In the local government scenario, our software agent requests a copy of its user's birth certificate from the local government. The local government delivers copies for a certain price. However, through some technical glitch the request for the copy arrives twice at the government offices. The local government accordingly sends two copies and charges the account of the user twice. Our user is far from pleased. Can he get (half of his) money back? Does it make a difference if the technical failure occurred in the software agent, in the computer system of the local government or in the system of the ISP that provided facilities? If the problem can be traced to the software agent, does it matter whether the user himself choose to use the software agent or that the government prescribed the use of the software agent?

5.1.2 Confidentiality

In the hospital scenario, software agents guard over the patient's dossiers, making sure that only authorised personnel obtains access to the sensitive data stored in these. The robustness of the system guarding access to the data is of importance. Is password protection sufficient or is biometric verification required? If a password is adequate, how often must it be renewed? And what procedure holds if a doctor claims to have lost his password?

5.1.3 Non-Excessiveness

In the local government scenario, our software agent contacts a software agent of the local government to collect information about building regulations. The information is available for free, but access is only granted if our software agent reveals the identity of its user. May the government agent request this information as an access condition? Does it matter for what purpose this information is asked? For a more efficient enforcement of building regulations or simply for selling it to banks (who like to know who might be interested in mortgages)?

5.2 RECIPE FOR RELIABILITY

The recipe for reliability is analysed from a legal perspective in Section 5.2.1 and a technical perspective in Section 5.2.2.

5.2.1 Legal Analysis

In Chapter 2, in the second recipe, the rules concerning the formation of contracts are explained. Contracts are formed by exchanging an offer and an acceptance of that offer. According to the subjective approach, a statement is binding if it correctly represents the will of the declaring party. However, if there is a discrepancy between the statement and the real intention of the declaring party, then the statement can *still* be binding if the party receiving the statement is allowed to trust the statement to be in accordance with the intention of the declaring party. The objective approach is "merely" concerned with the outward appearance of the statements that contracting parties exchange. This recipe focuses on the situation in which there is a miscommunication because a statement is delivered wrongly.

5.2.1.1 Communication

When dealing with Dutch law, which adheres to the subjective approach, this problem of miscommunication should be approached from the point of view that the declaration differs from the intention of the declaring party. There can be several causes for this discrepancy between declaration and intention:

- '*vis absoluta*', this occurs, e.g., when someone is physically forced to perform a certain action, like signing a statement;
- impairment of mental faculties, this can occur in case of drunkenness, narcosis, absent-mindedness or great excitement;
- slip of the pen or tongue;
- wrongly delivered statement by telegram or messenger (Hartkamp and Tillema, 1995; Hartkamp, 2001).

With respect to agents, a number of technical difficulties might occur because of which the agent makes statements that do no comply with the intentions of its user:

- the user might give the agent incorrect instructions, accidentally or otherwise;

- the agent might make incorrect statements, despite correct instructions from his user. The reason for this could lie in the workings of the agent itself, or, the method of communication that the agent uses might be faulty.

The main legal question that arises is: when is a person bound by the statement that the person uttered, even though the statement does not correspond with the internal will of that person? As discussed above, a person is not bound by his statement if the statement does not reflect the person's will. However, the statement can still be binding because of the *reliance theory*. This principle can be found in Article 3:35 of the Dutch Civil Code (DCC):

> The absence of intention in a declaration cannot be invoked against a person who had interpreted another's declaration or conduct, in conformity with the sense which he could reasonably attribute to it in the circumstances, as a declaration of a particular tenor made to him by that other person.

When dealing with the reliance theory, Article 3:11 DCC should be mentioned as well. Article 3:11 DCC describes a person's good faith:

> Where good faith of a person is required to produce a juridical effect, such person is not acting in good faith if he knew the facts or the law to which his good faith must relate or if, in the given circumstances, he should know them. Impossibility to inquire does not prevent the person, who had good reasons to be in doubt, from being considered as someone who should know the facts or the law.

Because of Article 3:11 DCC a party may have a duty to investigate whether the other party's statement depicts his true intention correctly.

To determine whether a party is allowed to rely on a statement made by the other party, the detriment suffered by the declaring party can play a role (Hartkamp, 2001, nr. 104); the declaring party may suffer (possibly financial) detriment because of his incorrect statement. If this detriment should be obvious to the party receiving the statement, then this might be a indication that this party ought to investigate whether the statement is in accordance with the intention of the declaring party.

As is obvious from the above, so far the exact technical cause for a miscommunication is not relevant for the determination whether a party can invoke the reliance theory. However, one can propose to allocate the risk of possible technical failures according to the rule that these risks should be borne by the person who is most likely to be able to prevent them from becoming reality. Using this rule, the following allocation can be made:

- risks concerning faults in the internal workings of an agent should be borne by the user of that agent;
- risks concerning faults in the communication should be borne by the person choosing the means of communication (compare Article 37(4) DCC);
- risks concerning faults at the receiving end of the communication should be borne by the recipient.

Please note that the allocation of these risks is only relevant for assessing the application of the reliance theory.

Subsequently, the question can be put as to which party should take precautions that could prevent these risks from happening. The risks could result from certain choices:

- the choice in technology (simple or complicated?);
- the quality of the technology that is used;
- the use that is made of the technology (for instance typos or incompatible software).

The accountability for the choices that are made can be allocated to respectively:

- the person that chose the technology;
- the person that supplies or maintains the technology;
- the person using the technology.

Whether the user mentioned in the local government scenario sketched above is remunerated depends on what went wrong and who has to bear the risk for that. If, e.g., the double sending is caused by a bug in the user's software agent, the user will generally bear the risk. This may, however, be otherwise if the local government provided the software agent and the user did not and reasonably could not discover the bug.

5.2.1.2 Legal requirements

This recipe deals with reliability in the communication between software agents. The law distributes the consequences in case of miscommunication. Technology is in a position to detect and prevent miscommunication.

Measures to be taken by producers of software agents:

- There must be ‘handshake protocol’ that enables two software agents who wish to communicate to determine whether they can ‘talk’ to each other. The handshake protocol must be ‘universal’ and here lies a role for standardisation and/or the law.
- There must be a 'handshake protocol' that enables two agent platforms to successfully migrate an agent from one to the other. Another issue for standardisation and/or the law.
- It must be made transparent to the user of a software agent that a software agent supports the ‘universal’ handshake protocol.
- The result of the handshake (communication is or is not possible) and the reason (what communication ‘language’ is chosen) must be logged.
- For high profile purposes, it is desirable to have not only syntactical compatibility, but also a (high and perhaps scaleable) degree of semantical compatibility.
- The authenticity, integrity and availability of messages exchanged must be warranted using the measures dealt with in the recipes of this chapter and the previous chapter.

Measures to be taken by the users of software agents:

- The user must make sure that his software agent is able to perform the ‘handshake process’.
- The user must retain the loggings that record the outcome of the handshake process.

5.2.2 Technical Analysis

Reliability for agents and agent platforms hinges on two aspects: reliability of communication and processing.

5.2.2.1 Reliable communication

Reliable communication plays a role among agents, but also between agents and agent platforms: in both situations information needs to be transferred in a form which expresses the sender's intent and is understandable by the receiver. In the local government scenario, a request for a copy of a user's birth certificate is communicated from the user's agent to an agent from the local government: the user's agent needs to be certain that the request arrived, remained unchanged, and could be understood by the recipient. The recipient may provide feedback on receipt of the message, possibly including details about the contents, thereby signifying understanding. Techniques and issues are briefly discussed below.

Communication languages prescribe protocols, languages and ontologies to be used when communicating. Agent communication languages (ACLs), such as KQML and FIPA-ACL, commonly include in the message a reference to the language and ontology used (Labrou, Finin and Peng, 1999). ACLs and protocols commonly have formal semantics, which expresses how a message is to influence the (mental) state of the recipient. Verification whether an agent's implementation conforms to these semantics is not easily achieved nor determined.

The semantic interpretation of the content of the message, e.g., whether the concept 'bier' refers to 'beer in Dutch' or 'big stone slab', is left open in an ACL. Agents are programmed to use a specific ontology (possibly with formal semantics), or need to negotiate which ontology is to be used. The latter is not yet common, although the advent of the Semantic Web (Berners-Lee, Hendler and Lassila, 2001) facilitates reasoning about ontologies. No meta-protocol exists as yet, via which agents (and agent platforms) can negotiate which protocols, languages, and ontologies to use. Often, agents use directory services to announce their communication abilities.

Standards are lacking, and although FIPA-ACL has become more widely used in the AgentCities project (Willmott, Dale, Burg, Charlton and O'brien, 2001), more research is clearly required, especially concerning the role of ACLs in open systems. Standards do not provide the solution, as when two agents use the same protocols, languages and ontologies, interoperability is not guaranteed: they may interpret the same message in a different way.

In addition to communicating among agents, agents also communicate with agent platforms: to request access, to migrate to other agent platforms, etc. Agent platforms themselves also communicate with other agent platforms (and other services such as location services) to migrate agents, exchange reputation based information etc. This communication also needs to be regulated and standardised, as no standards exist (FIPA's proposals are not finished nor adequate for open heterogeneous systems) nor meta-protocols.

The use of the same protocols, languages, and ontologies does not imply that the involved agents and agent platforms are able to correctly construct and interpret messages. It is important to note that agents and agent platforms employ multiple protocols, languages and ontologies, with different semantics. Agents and agent platforms are equipped with 'matching' and 'transformation' procedures, to translate between different protocols, languages and ontologies, which may be imprecise and erratic. Current research on the Semantic Web includes

research on 'ontology-mapping' problems. Logging communication by agents and agent platforms, including their intent or interpretation requires extensive storage capacity for detailed traceability logs, whose correctness may not be proven nor be usable for legal purposes.

Agent platforms are to provide adequate techniques to ensure reliable communication across the Internet: a notably unreliable network (Tanenbaum and Steen, 2002). Most communication processes offered include a better quality of service, including:

- reliable: a message is guaranteed to arrive at its destination (possibly with an upper bound on the time taken in transfer),
- causal-ordered: two messages send after another are received in that order,
- secure: a message is transferred while maintaining confidentiality and integrity.

5.2.2.2 Reliable processing

An agent's processing reliability depends on two factors: the agent's code and the agent's environment. In the local government scenario, both the reliability of the agents, as well as their environment needs to be ascertained. Careful design and development of the agents and the environment may yield more reliable processing, yet other mechanisms need to be in place as well, including trust, e.g., in (types of) agents used by civilians.

An agent's code may be trusted to a certain extent, based on reputation of the agent's code (e.g., reputation of other agents with the same code and reputation of the producer of the code) and analysis of code, e.g., by formal verification techniques (a costly endeavour).

More important is an agent platform which hosts an agent: is the agent platform trustworthy? Does an agent platform provide the needed resources to the agent? Is a stable execution environment provided? Is an agent actually 'started', or merely discarded? Is the agent platform capable of reliable migration: if an agent tries to migrate to this agent platform, but is rejected, is it then returned to its previous agent platform? Protocols may be used, e.g., by providing 'contracts' or 'leases' (e.g., certificate based) which bind an agent platform and an agent to a certain extent. Trustbuilder (Winslett, Yu, Seamons, Hess, Jacobson, Jarvis, Smith and Yu, 2002) employs such an approach to regulate access control: policies and certificates are exchanged until either a satisfactory solution is found, or the agent cannot adhere to policies and seeks access elsewhere.

For more information on agent platform's processing reliability, see Section 4.2.2.3 about agent platforms' integrity.

5.3 RECIPE FOR CONFIDENTIALITY

The legal analysis describes statutory provisions that deal with the confidentiality of data. Three categories of provisions are addressed: 1: criminal law provisions that penalise the impairment of (the confidentiality of) computer data; 2: data protection provisions that protect personal data; and 3: provisions that impose confidentiality obligations on the holders of certain information, for instance lawyers or doctors.

5.3.1 Legal Analysis

The Dutch Criminal Code contains a number of provisions that deal with computer related crimes, which can also include the protection of confidentiality of data. There are three principles that underlie these provisions. The three principles relate to the interests that must be protected by the provisions. The principles are:

1. availability;
2. integrity; and,
3. exclusivity (Franken and Kaspersen, 2001).

All three principles can relate to both hardware and data (including software). The third principle, exclusivity, can also entail the *confidentiality of data*, which is the central principle of this recipe. However, when reading the criminal law provisions, one must keep in mind that the provisions can entail more than "just" the protection of the confidentiality of data.

5.3.1.1 Definitions

Two important definitions in the Dutch Criminal Code are those concerning 'data' and 'computerised devices and systems'. Article 80quinquies defines 'data' as any representation of facts, concepts or instructions, whether agreed upon or not, suitable for transfer, interpretation or processing by humans or computerised devices and

systems. According to Article 80sexies, 'computerised devices and systems' means a facility for the purpose of storing and processing data by electronic means.

5.3.1.2 Interference with computer systems

As to the protection of the confidentiality of data, a number of provisions of the Dutch Criminal Code might be relevant. First of all, Article 161sexies penalises the interference with computer systems. According to the article 'a person who intentionally destroys , damages or renders unusable any computerised device or system for storing or processing data or any telecommunications facility, or who intentionally causes the defective functioning or operation of such device, installation or facility or intentionally frustrates a safety measure taken with respect to such device' is criminally liable. Article 161septies penalises the same act when the person committing the act does so due to his negligence or carelessness instead of acting intentionally. Obviously, the main focus of the Articles 161sexies and –septies is not to protect the confidentiality of data, but it is not impossible that the interference with computer systems compromises this confidentiality.

5.3.1.3 Computer intrusion

Article 138a deals with the intrusion into a computer system. According to this article, a person who intentionally unlawfully intrudes into a computerised device or system for storing or processing data or a part of such device or system, is guilty of computer intrusion if a: the individual thereby breaches any security, or b: gains access by technological means, with the help of false signals or a false key, or by assuming a false capacity. These last two criteria make it clear that a computer system cannot be allowed to operate without some kind of technical measures that protect the access to the computer system. Furthermore, this protection must be more than just a symbolic barrier: the door must not only be closed, it must be locked as well (Franken and Kaspersen, 2001, p. 399). In the legal field is discussion about the tenability of the requirement of 'breach of security'. The European Commission tinkers with the thought to abolish this requirement. An individual who is guilty of computer intrusion faces a heavier punishment if he subsequently copies the data stored in the computerised device or system to which he has gained access unlawfully, and records such data for his own use or that of another. Here, the confidentiality of the data is addressed explicitly. Additionally, computer intrusion through the telecommunications infrastructure or a telecommunications facility used to service the general public is punished more heavily if the offender subsequently uses processing capacity of a computerised device or system with the object of obtaining unlawful gain for himself.

5.3.1.4 Interception of data transmissions

Article 138a, described above, penalises the intrusion into a computer system. In some cases this can compromise the confidentiality of data. Apart from this situation, there is a number of instances which relate more directly to the confidentiality of data. In these cases the interception of data transmissions is penalised. First of all, according to Article 139a of the Dutch Criminal Code, it is punishable to intentionally intercept or record data that are being transferred in a dwelling, enclosed room or premises by means of a computerised device or system. Secondly, according to Article 139b, the interception of a data transmission outside of a dwelling, enclosed room or premises is also punishable. Thirdly, according to Article 139c, a person is punishable when he, with the help of a technical device, intentionally taps or records data that is not intended for himself or for the person by whose order he is acting. Article 139c deals specifically with data that is transferred by means of the telecommunications infrastructure or through a telecommunications facility that is used to service the general public, or by means of the peripheral equipment connected to it.

5.3.1.5 Destruction of computer data

Finally, Article 350a can be mentioned, although the main purpose of this article is to protect the availability and the integrity of data and not so much the confidentiality of the data. Article 350a penalises the destruction of computer data. According to the article, a person who intentionally and unlawfully alters, erases, renders unusable or inaccessible data that is stored, processed or transferred by means of a computerised device or system, or adds data thereto, is criminally liable. Subparagraph 3 of Article 350a deals with the use of virus-like programs. According to subparagraph 3, a person who intentionally and unlawfully makes available or disseminates data that are intended to cause damage by replicating in a computerised device or system is criminally liable. Subparagraph 2 of Article 350b contains a similar rule, but merely requires that the person acted negligent or careless.

5.3.1.6 Holders of confidential information

The Dutch Criminal Code contains two provisions that relate to duties to hold certain information confidential. Article 272 deals with certain holders of confidential information. Article 273 deals with confidential corporate information. According to Article 272 a person by whom any secret which he either knows or should reasonably suspect that he is bound to keep by reason of his office, profession or a legal requirement, or his former office or profession, is intentionally violated is punishable. The doctor mentioned in the scenario at the beginning of this chapter provides an example. He may not only disclose information about his patients, he also must make sure that the information is not revealed through the software agents he uses. Article 273 states that a person is punishable if he intentionally: (1) discloses specific information related to a commercial, industrial or service organisation in which he is or has been employed, which he was bound to keep secret or (2) discloses, or uses for motives of pecuniary gain, data that have been obtained by means of a criminal offence from a computerised device or system of a commercial, industrial or service organisation and that are related to such organisation, where the data, at the time of disclosure or use, were not generally known and where any disadvantage may ensue from such disclosure or use.

5.3.1.7 Legal requirements

Measures to be taken by the user of a software agent:

- The user must identify what information needs to remain confidential and who may have access to the confidential data.
- The user must take adequate measure to protect the confidentiality. Thereto, he must decide whether it suffices to protect the data by regulating access to the hardware where the data are stored; this may be the case with a static agent. Alternately, it can be required that the data themselves are protected, e.g., through encryption.

Measures to be taken by government or third parties:

- Confidential data may generally only be revealed to specific persons or machines. Adequate protection of confidentiality, presupposes the existence of a scheme whereby the intended recipients can 'prove' their identities or at least show their authorisation to 'read' the confidential data.
- A marker could be agreed upon that indicates that the associated data are 'confidential'. Through some form of government or self-regulation, 'respect for the marker' could be made mandatory.

Measures to be taken by producers of software agents:

- Mobile software agents must support confidentiality, by making encryption of their data load possible.

Measures to be taken by the holder of an agent platform:

- The holder of an agent platform states his policy with respect to the confidentiality of data contained in software agents. This policy involves at least the following: (1) under what conditions the holder may or may not read the data and (2) what measures he has taken to prevent 'reading' by third parties.

5.3.2 Technical Aspects

Confidentiality, a.k.a. privacy of agents, or protection of data contained by agents, is subject of current (legal) research, e.g., (Borking, van Eck and Siepel, 1999; Bassoli, 2002; Villecco Bettelli, 2002). Confidentiality is mainly about protecting secrets (Anderson, 2001), i.e., disclosing information only to authorised parties (Tanenbaum and Steen, 2002). Techniques for ensuring confidentiality are described in Sections 3.2 to 3.4 (recipes for Identity, Identity Protection, and Anonymity), including cryptographic techniques. Access control mechanisms are most important to protecting information (including the agent or agent platform) from unwanted modifications by other agents and agent platforms, e.g., as provided by Trustbuilder (Winslett, Yu, Seamons, Hess, Jacobson, Jarvis, Smith and Yu, 2002). See Section 4.2.2 for issues and problems with maintaining integrity of data, agents (including agent migration) and agent platforms, which are similar to maintaining confidentiality.

The role of protocols and traceability for maintaining confidentiality and regulating access is of paramount importance,. Standardisation requires more research on these topics.

It is important to note that trust and confidentiality are closely related: when one agent discloses confidential information to another agent, the other agent is in general trusted to not disclose this information to anyone else. In sum, confidentiality is always temporary, and minimal confidential information should be placed in agents and agent platforms.

In case of applications involving confidential user data adequate techniques need to be deployed to assure confidentiality. Tracing mechanisms need to be in place to monitor possible mis-use. Open systems such as the chemical commodities marketplace are less easy to control than closed systems such as the hospital scenario. In both systems, however, contingency plans need to be made in case breaches in confidentiality are detected.

5.4 RECIPE FOR NON-EXCESSIVENESS

One of the principles of informational privacy is that not 'too much' personal data must be collected or processed. This may become problematic with respect to software agents, since they are generally not able to distinguish between personal data and other data.

5.4.1 Legal Analysis

Non-excessiveness plays an important role in informational privacy. The processing of personal data is namely governed by a number of principles. With respect to non-excessiveness, three principles are especially relevant: the purpose specification principle, the collection limitation principle and the use limitation principle. The former indicates that the purpose for which personal data are collected and processed must be specified in advance. The latter two principles entail that the collection and processing of the personal data must not be incompatible with the specified purpose. Directive 95/46/EC expresses these principles as follows[6]:

Recital 28

28. Whereas any processing of personal data must be lawful and fair to the individuals concerned; whereas, in particular, the data must be adequate, relevant and not excessive in relation to the purposes for which they are processed; whereas such purposes must be explicit and legitimate and must be determined at the time of collection of the data; whereas the purposes of processing further to collection shall not be incompatible with the purposes as they were originally specified;

Article 6

1. Member States shall provide that personal data must be:
    a. processed fairly and lawfully;
    b. collected for specified, explicit and legitimate purposes and not further processed in a way incompatible with those purposes. Further processing of data for historical, statistical or scientific purposes shall not be considered as incompatible provided that Member States provide appropriate safeguards;
    c. adequate, relevant and not excessive in relation to the purposes for which they are collected and/or further processed;
    d. accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that data which are inaccurate or incomplete, having regard to the purposes for which they were collected or for which they are further processed, are erased or rectified;
    e. kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the data were collected or for which they are further processed. Member States shall lay down appropriate safeguards for personal data stored for longer periods for historical, statistical or scientific use.

The principles have found their way to the Dutch Data Protection Act (hereinafter: DDPA), i.e., the implementation of Directive 95/46/EC. In unofficial translation, the pertinent provision regarding the limitation principles reads a follows:

Article 11 DDPA

1. Personal data are only processed if and to the degree that they are adequate, relevant and not excessive, taking account of the purposes for which they are collected and subsequently processed.

N.B. the provision only talks about processing, but collecting is also a form of processing.

[6] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, Official Journal L 281 , 23/11/1995 P. 0031 - 0050

5.4.1.1 Collection of personal data

One of the grounds for collection and processing of personal data without the unambiguous consent of the data-subject is the following: the data may be processed, if the processing is necessary for the purposes of the legitimate interests pursued by the controller or by the third party or parties to whom the data are disclosed, except where such interests are overridden by the interests for fundamental rights and freedoms of the data subject. A software agent of an Internet Access Provider may, e.g., want to register what websites the subscribers of the access provider often visit. Often visited sites may then be placed in the proxy-cache of the access-provider in order to make efficient use of bandwidth. For this purpose, it is not allowed to register which users visited which websites. For the intended purpose, it is enough to register quantitative information with respect to websites. Registration of subscribers thus has to be left behind (Esch, 2002).

By the same token a local government in the scenario-example mentioned earlier, may only collect identity-information if it has a legitimate interest in doing so. This could be the case if the local government can show that enforcement of building regulation will be much more effective, having the identity data at its disposal.

5.4.1.2 Processing of personal data

The processing of personal data is limited to specified and justified purposes. The software agent of an access provider could, e.g., collect and process data about the uses a subscriber makes of his services for billing purposes or for advising the subscriber about the best equipment (modems, cable etc.) to use given his use of services. However, making a user profile on the basis of the same data for marketing purposes may very well be outside the specified purposes and must be left behind, if no additional measures are taken, such as asking for consent.

5.4.1.3 Termination of processing

If the software agent processes personal data, it must be clear for what duration this information is needed for the functioning of the software agent. Once identificatory data become superfluous (e.g., because the task is completed) the identificatory information has to be removed or must be placed under a special regime (Horst, 2002, p. 107). An example may clarify this: if a commercial transaction has been performed and all possible legal claims are precluded by the lapse of time, all identificatory information relating to the transaction has to be removed from its storage place or it has to be made anonymous.

5.4.1.4 Legal requirements

The user of the software agent:

- The user of the software agent must make clear for what purposes the software agent may collect and process personal data.

The producer of a software agent:

- The autonomy of a software agent may not reach so far that the software agent collects or processes personal data without its user being able to foresee such collection or processing.
- The software agent must support the removal or anonymisation of personal data once they can no longer serve their purpose.

5.4.2 Technical Analysis

Enforcing non-excessiveness is difficult for software in general, and agents and agent platforms in specific. Technically, certificates may be used to explicate intent, protocols to regulate interaction, and traceability to provide evidence in case of complaints, although it is difficult to detect excessiveness. However, no guarantees can be given, for both agent platforms and agents: trust plays an important role.

In the hospital scenario, agents from insurance companies need to acquire data from the hospital's patient records for statistical and financial purposes (e.g., to pay for treatments). However, a patient's right to (medical) privacy needs to be guaranteed. The insurance company's agents, however, need to acquire data about individual patients and groups of patients: a situation in which easily too much private data about individual patients may be gathered. Stringent measures are needed to control access to data by insurance agents, and transfer of amalgamated data to insurance companies. Policies need to be in place to deter, and respond, to data gathering excessiveness by insurance agents, including killing agents, banning all insurance company's agents, etc (see Section 4.2, Integrity, for a discussion of additional measures against malicious agents).

5.4.2.1 Non-excessiveness for agent platforms

Agent platforms need to collect and process (monitoring) information about local agents, other agent platforms and remote agents, e.g., to assess risks, optimise local performance and reliability, and pro-actively respond to changes in their environment. Protocols are needed which specify which data may be stored for which period of time. Note that an agent platform, which is akin to an Internet Service Provider (ISP), may be (legally) required to maintain certain tracing data. Additional issues are similar to issues encountered by enforcing non-excessiveness for agents.

5.4.2.2 Non-excessiveness for agents

Agent platforms may collect and process large volumes of information. It is difficult for producers of agents to foresee whether such information will concern personal data (recognition problem), let alone for users of agents. Anonymisation or removal of personal data is only possible if that data can be identified.

Potential problems for agents include:

- Agents may employ learning algorithms: (which is quite common!), on the basis of which it is unforeseeable what information the agent needs and uses in the future.
- Agents should minimally store sensitive information in themselves, but may send this to other agents or services somewhere on the Internet, place it in a (distributed) data object for later referral, encode this information (see text on encryption in Section 3.2.2), etc.
- Any 'detection module' placed in an agent can probably be circumvented (see the remarks about enforcing traceability or evidence gathering).

Potential technical solutions:

- Have inspectable agent data (which resides in agent container, in a standardised format).
- Trace all information entering, and leaving the agent (which may be encrypted): may yield enormous traffic analysis data, if content is also stored: infeasible.
- Severely limit the type and amount of data an agent may process (infeasible: agent may need to look through many websites to find something, possibly downloading megabytes or terabytes of data).
- Severely limit the amount of data an agent may transport during its migration (not feasible: how to set the limit, how to prevent agents from using smart compression algorithms, …)
- …

Potential social solutions:

- Avoid placing identificatory information in agents.
- Provide minimal information to other agents/parties.
- Employ minimal helper agents.
- Use pseudonyms (e.g., an identity which can easily be removed/discarded, which is not a human's *real* identity) which are frequently discarded, thereby hoping to thwart information gathering by other parties, and linking this information to a human's identity. Cf. throw-away web-based email accounts.
- Build webs of trust to help in choosing between agents and agent platforms.
- …

6 DISCUSSION

The main objective of the ALIAS project has been to explore the legal implications of the use of software agents, providing guidelines to software developers for agent design and implementation. The exploratory nature of the project provided an opportunity to study a range of related topics, discovering numerous legal considerations. Six key intermediary concepts have been identified: autonomy, identifiability, traceability, integrity, originality and trust. The first five are concepts that have specific associated legal and technical equivalents. Note that aspects related to software development in general have not been included in this report. This list is not exhaustive, nor are the descriptions of these concepts, and the legal and technical requirements identified. They do, however, together provide a frame of reference with which the field can be further studied. Autonomy is the subject of Chapter 2. Identifiability and traceability are discussed together in Chapter 3, integrity and originality in Chapter 4. The last concept, trust, the subject of Chapter 5, is a social concept that is strongly related to the other concepts.

6.1 OPEN AND CLOSED SYSTEMS

The intermediary concepts identified in this document have been studied both within the context of open and of closed systems. The level of regulation differs: within a closed system the actors are known, data, procedures and actions are mostly well-defined and can be verified. This implies a high level of integrity: it is clear who can do what, when and with which information. Hospitals are examples of closed systems. Procedures define which patient data, for example, may be used for which purpose, how it can be used and how it can be obtained. Another clear distinction between open and closed systems is the distinction between the Internet (open) and an intranet (closed). There are many examples of legal provisions in which legal consequences are dependent upon the openness and closedness of access to data, or (human or computer) actions or systems. The criminalisation of hacking protects, e.g., systems and all data and processes within their perimeter against an attack by unauthorised parties (Art. 138a Dutch Criminal Code). Telecommunications operators have to protect their systems against eavesdroppers (Art. 11.3 Telecommunications Act). Other provisions are more closely tailored to the protection of data themselves (instead of protection of the system they reside in). The Dutch Data Protection Act demands, e.g., that personal data are suitably protected against loss or unlawful processing (Art. 13). Other legal rules do not mention the concept of data, but may nonetheless have far-reaching implications for data. For example, a medical doctor has to abide to his professional confidentiality (Art. 272). This implies that he may not orally disclose information about the medical status of a patient; it also means that he has to take sufficient security measures to ensure that patient data stored in his computer system remain confidential.

From a technical perspective, important features of open and closed systems are: the data in the system, the procedures that can be attributed to the data, and the possible actions on these procedures and data, also called tasks. With these three basic features, a closed system requires a model that enforces access control on the data, procedures, and actions that are used by an agent in an organisation. In other words, there are restrictions on the information flows implemented with and within such an organisation. But as the actors are known, the model can regulate the access of specific agents to specific data and procedures. This implies that not all concepts are equally important for agents in closed systems. For instance the fear that agents will be affected by a hostile agent platform, is much less present in closed systems than in open systems. As a result, the requirements dictated from a legal point of view may vary according to the task of the agent and the environment in which this task is carried out.

The distinction between open and closed systems is of importance. Protocols for closed systems can be highly detailed, as the actors and tasks are known in advance. The protocols can often be well specified. For open systems standards, both for agents and for agent platforms, may be the only means to regulate interaction. It remains to be seen if protocols and standards will suffice: whether concepts such as good faith can be captured by such means.

The legal implications of the use of software agents in both closed and open systems are yet not fully understood. It is often unclear if existing legislation is applicable, and when. General legislation with respect to the use of software is available (e.g. copyright law, contractual law and general provisions under civil law), but there is little tradition, and there still are few accepted procedures, with respect to the specific use of software agents. Such general legislation, including directives for e-commerce and distance selling, are based on the concept of human intervention. Agents, however, do not always need to rely on human intervention. An important characteristic of argents is that they can act autonomously. An additional complication is the cross-

border activities in which agents engage. It is often unclear which national law applies in specific situations and under what conditions.

6.2 BALANCING INTERESTS

As mentioned above, one of the conclusions of this study is that in considering the interaction between legal requirements and agent design, several new balancing choices have to be made and decisions have to be taken. In other words, the introduction of agent technology means that other considerations must be dealt with when balancing well-known interests. For example, software agents can be instruments in both hiding and uncovering identities. They may also inadvertently play a role in the sense that they leave traces that may be helpful to somebody wishing to uncover the identity of the user of the software agent. Thus, an agent may play a role in sharing as well as shielding information. When applying this to the legal setting, the law requires or allows information to be shared in certain situations, whereas in other situations information must be shielded. This means that agent designers should be aware that the agents developed should facilitate balancing of interests (i.e., between hiding and shielding information) when such balancing is required.

This example shows that in the interaction between legal requirements and agent design, several choices and decisions have to be made on issues that reflect a tension between two interests. When looking at the previous chapters, the following tensions can be identified:

- Tension 1: the tension between sharing and shielding information by a software agent (for example knowing or not-knowing the identity of the owner of an agent);
- Tension 2: the tension between control and dependency;
- Tension 3: the tension between freedom (to pursue ones own interest) and obligation (to take account of other player's interests).

All of these tensions can occur in a number of relations: the relation between (the producer of) a software agent and its user, the relation between software agents or their users[7] and the relation between a third party (e.g. the manager of an agent platform) and the users of a software agent. All three tensions can be illustrated by means of the concepts discussed in the Chapters 2 to 5.

6.2.1 Tension 1 - Sharing and Shielding Information

When it comes to the concept of autonomy (discussed in Chapter 2) the tension between sharing and shielding information is primarily seen in the relation between a software agent and its user. From a legal perspective, there is a strong argument in attributing the acts of a software agent to its user. This would mean that transparency between a user and his or her software agent is of utmost importance, meaning that information (for example on the identity of the user of the agent) cannot be shielded. Also, the user must be able to express what he expects of the software agent, the software agent must make clear to its user what it can and may do. The latter must, however, not lead to an information overload of the user (too much information is no information).

As regards the concept discussed in Chapter 3, identifiability and traceability, note that in networked on--line environments, diverging needs exist. Software agents can be instruments in both hiding and uncovering identities. Software agents may also inadvertently play a role in the sense that they leave traces that may be helpful to somebody wishing to uncover the identity of the user of the software agent. When considering the tension between sharing and shielding information, it is clear that a software agent is certainly able to have an important role in shielding identity-information. There is, however, a more intricate aspect to knowledge in relation anonymity. There is no sense in shielding identity-information if the information is not to be shared under certain conditions. The best encryption technologies are to no avail if a software agent is defective in determining under which conditions what information can be shared with what other parties. Protection of anonymity comes to depend upon knowledge about others and interpretation of situations in which a software might find itself.

As regards the concept of integrity and originality (discussed in Chapter 4) the tension is reflected in that certain knowledge must be available as regards the proper functioning of a software agent. Chapter 4 mentions that the proper functioning of software agents is highly dependent upon the conservation of their integrity and the integrity of the platforms on which they function. Integrity means here that no data are unduly altered, erased or supplemented and that the physical objects involved (such as computer systems are not damaged or destroyed). Originality concerns the ability to distinguish between originals and copies. Legally, originality is *inter alia*

[7] ·Deserve middle agents and end agents a separate category?

required for ensuring uniqueness; this, e.g., is the case with bills of lading. The above means that certain knowledge is of utmost importance and that shielding such information is not preferable.

Finally, as regards the concept of trust (discussed in Chapter 5), the tension between sharing and shielding information is reflected for example in confidentiality and non-excessiveness. Confidentiality presupposes that it is possible to distinguish between persons, systems and processes that may have access to information and those that do not. Hence, the availability of related information is crucial. Non-excessiveness requires that a software agent that collects personal information knows for what purposes this information is needed and how long it is needed. Thus, a software agent must have such information and understand that this information is personal information. Shielding such information is thus not to be preferred.

6.2.2 Tension 2 - Control and Dependency

In looking at the concept of autonomy (discussed in Chapter 2) note that the tension between control and dependency plays a role with respect to imputability. In view of the imputability of the acts of a software agent to its user, the latter must exert control over his software agent. The software agent must make clear what it does and must always allow the user to pull the plug. The question is whether control by the user is sufficient for keeping a software agent in check. By the application of protocols to which not only our user's software agents abides, but also other agents, software agents are much better to control than would be possible by unilateral efforts of the user himself. It makes the user, however, more dependent upon protocols (is there a suitable protocol, is it unambiguous? Do other software agent or platforms adhere to the protocol?).

The tension between control and dependency is also reflected in the concept discussed in Chapter 3 (identifiability and traceability). As discussed, ones informational privacy is in networked on-line environments under constant threat. Anonymity is often hailed as the means to bring back control over 'informational privacy' to the person concerned. In this sense, anonymity is a matter between a user and its software agent; the latter shields the identity of its master. However, not saying ones name may be not enough to remain anonymous. Ones software agent's behaviour on line could be monitored (think, e.g., of traffic data, or navigation through a website); small pieces of information about this behaviour could be collected that are as many small clues to ones identity. In the end, it appears that anonymity comes to depend on third parties (ISPs, owners of agent platforms, website owners, etc.) refraining from piecing an identity together. This underlines the necessity of protocols for maintaining anonymity on line.

As regards the concept of integrity and originality (Chapter 4) note that in view of the interest of integrity, parties are dependent upon each other. A mobile software agent process that finds itself on a platform is often dependent upon that platform for maintaining its user's anonymity. As regards originality, it is clear that digital data depends upon an infrastructure. Originality of digital data is non-existent or unusable, and must be realised by the infrastructure in which the original data are to be used.

Trust, as discussed in Chapter 5, also reflects the tension between control and dependency. For example, the law places the preponderance of responsibility for 'correct' declarations with the user of the software agent. This means that the user must be able to exert sufficient control over the software agent. The existence of a 'trade language' for software agents can take away many sources of miscommunication. This introduces dependency upon such a 'language'. Also, control is a crucial issue in confidentiality and availability. Confidentiality is difficult to maintain absent any protocols. With respect to availability, the non-repudiation of receipt only seems to be possible with the co-operation of the receiver. Finally, as regards non-excessiveness it is clear that a software agent must be able to destroy information when it is no longer needed.

6.2.3 Tension 3 - Freedom and Obligations

With respect to autonomy (Chapter 2), it transpires that software agents allow a user to take better care of his interests than would be possible without them. Many parallel negotiations may, e.g., be opened. The question is how the growing possibilities for pursuing ones self-interest can be counterbalanced by the responsibility for the interests of others that could be the victim of such activities?

As regards the issues discussed in Chapter 3 on identifiability and traceability, note that the tension of freedom and obligations is reflected in the fact that, for example, when a user desires its agent to act anonymously, various interests (and thus the law) require that the agent identifies itself. Hence, there is an obligation to identify (and no freedom to remain anonymous)

With respect to integrity and originality (Chapter 4) it is clear that freedom and obligations play a role in determining in what way the different parties must take account of the interests of the counterparts. On the one

hand, the justified interest in maintaining his systems integrity may mean that a platform owner may impose certain restrictions upon software agents wanting to enter his system. On the other hand, the interest of the software agent's user could require that a platform owner may not always terminate or slow down agent processes.

Finally, as regards the concept of trust as discussed in Chapter 5, balancing freedom and obligations could imply, for example, that the law imposes obligations to send confirmations on certain acts in networked on-line environments (for example a confirmation of a receipt of an acceptance to order a certain product). Also, as regards the issue of non-excessiveness, the legislator has imposed duties in order tot protect the informational privacy of persons.

6.3 CONCLUSIONS

This project has indicated many areas in which more, deeper research is needed, especially with respect to the legal status of agents, the legal implications and normative conditions to be applied to the use of agents. Actions of agents can be subject to many detailed legal requirements applicable to specific situations. The context in which an agent is used, and specific characteristics of the situation determine the requirements set from a legal point of view. It has become clear that some requirements are context dependent, especially related to open and closed systems. Also, the analyses has shown that although various general legal requirements that apply (such as (pre-contractual) information duties), several new dimensions on issues such as contract negotiation, contract closing, migration and acceptance of conditions for hosting, data access and interoperability arise. Sometimes, these new dimensions give rise to new regulatory questions, dilemmas and new dimensions in balancing well-known interests. An example of such balancing is the need for identification versus excessiveness. Many websites, for example, request identification before access is provided. This may not be in line with regulations with respect to excessiveness. How does this relate to agents requesting hosting on sites? We will discuss the conclusion in more detail underneath in the next section.

This document has focussed on agents in current practice. New possibilities that arise due to the status of agents have not been explored. For example, in a closed environment such as a hospital, privacy of patients is of great importance. Can agents be used to access and combine different sources of information if they can guarantee that the names of individual patients will never be considered/stored/associated with the data?

This document has also not solved the problems identified. With respect to security, for example, a number of research areas and sub-solutions have been proposed. Although confidential data can be encrypted, the problem of associating a private key with an agent is not trivial. An agent cannot securely carry its own private key. Agents may carry certificates; do these not need to be protected? Trusted third parties may be able to play an important role in solving these problems but the feasibility of such solutions needs to be further researched. Trusted third parties may also be of use for storing traces of agents as they traverse the Internet. Again the cost and feasibility of such solutions need to be studied. It does, however, seem clear that for the purpose of traceability a unique global identifier is needed. Whether this identifier needs to be known to all other parties is questionable; local identifiers may suffice.

Finally, in considering all that has been said in this document, we may conclude that the current state of agent technology and legal insight, makes it possible to use agents safely in closed and regulated environments. However, this does not mean that the use of agents, in particular in open systems such as the Internet, is not without problems. Traceability, certification, interoperability, protocols for negotiation, liability, technical instruments for insights in applicable law etc. are still not fully guaranteed. Hence, both technology and legal insight still need to be further acquired. This also holds for web services, which to some extent are comparable in this respect. In sum, the specific risks and obligations of the use of agents need to be further explored as well as their impact on the design of agents (and visa versa).

REFERENCES

Abdul-Rahman, A. and Hailes, S. (2000). Supporting Trust in Virtual Communities, *in Proceedings 33rd Hawaii International Conference on System Sciences*, pp. 9, IEEE Press.

Aberer, K. and Despotovic, Z. (2001). Managing Trust in a Peer-2-Peer Information System. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, pp. 310-317, ACM Press.

Alonso, E., D'Inverno, M., Kudenko, D., Luck, M. and Noble, J. (2001), Learning In Multi-Agent Systems, *Knowledge Engineering Review*, **16**(3):277–284.

Anderson, R. (2001), *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley Computer Publishing, New York.

Apistola, M., Brazier, F.M.T., Kubbe, O., Oskamp, A., Prins, J.E.J., Schellekens, M.H.M. and Voulon, M.B. (2002), Migrating agents: Do sysadmins have a license to kill? *in proceedings of the 3rd International SANE Conference*. pp. 399-401.

Apistola, M., Brazier, F.M.T., Kubbe, O., Oskamp, A., Schellekens, M.H.M. and Voulon, M.B. (2002a), Legal aspects of agent technology, *in Proceedings of the 17th BILETA Annual Conference*, pp. 11.

Apistola, M., Brazier, F.M.T., Kubbe, O., Oskamp, A., Schellekens, M.H.M. and Voulon, M.B. (2002b), Legal aspects of agent technology, *in* Blockeel, H. and Denecker, M. (eds), *Proceedings of The 14th Belgian-Dutch Conference on Artificial Intelligence (BNAIC2002)*, pp. 399-400.

Barber, K.S. and Kim, J. (2001), Belief Revision Process Based on Trust: Agents Evaluating Reputation of Information Sources, *in* Falcone, R., Singh, M.P. and Tan, Y-H. (eds), *Trust in Cyber-societies, Integrating the Human and Artificial Perspectives*, Lecture Notes in Computer Science, **2246**, pp. 73-82, Springer-Verlag.

Bassoli, E. (2002), Intelligent Agents and Privacy, *in* Sartor, G. (ed), *Proceedings of the workshop on the law and electronic agents, LEA2002*, pp. 45-51.

Bell, D.E. and LaPadula, L.J. (1973). *Secure Computer Systems: Mathematical Foundations and Model*, The MITRE Corporation, report MTR 2547 v2.

Bellare, M., Garay, J.A., Hauser, R., Herzberg, A., Krawczyk, H., Steiner, M., Tsudik, G. and Waidner, M. (1995), *iKP - A Family of Secure Electronic Payment Protocols*, IBM

Berners-Lee, T., Hendler, J. and Lassila, O. (2001), The semanticWeb. *Scientific American*, May, pp. 28-37.

Beth, T., Borcherding, M. and Klein, B. (1994). Valuation of Trust in Open Networks, *in Proc. 3rd European Symposium on Research in Computer Security*, pp. 3-18.

Bichler, M., Segev, A. and Zhao, J.L. (1998), Component-based E-commerce: Assessment of Current Practices and Future Directions, *SIGMOD*, **27**(4): 7-14.

Birk, A. (2001). Learning to trust, *in* R. Falcone, R., Singh, M., and Tan, Y-H. (eds), *Trust in cyber-societies : integrating the human and artificial perspectives*, Lecture Notes in Computer Science, 2246, pp. 133-144, Springer.

Bonabeau, E. and Theraulaz, G., (2000), Swarm Smarts, *Scientific American*, **282**(3):72-79.

Borking, J.J., van Eck, B.M.A. and Siepel, P. (1999), *Intelligent software agents and privacy*, Den Haag: Registratiekamer 1999, Achtergrondstudies en Verkenningen 13.

Boulmakoul, A and Sallé, M. (2002), Integrated Contract Management, *Proceedings of the 9th workshop of HP Openview University Association*, paper 4.1.

Bradshaw, J. M. (1997), editor, *Software Agents*. AAAI Press / MIT Press, Menlo Park, CA.

Brazier, F.M.T. and Oskamp, A.J. (2002), Agents: Nomads, Migrants or Globetrotters?, April, *Invited talk at the 17th BILETA annual conference*.

Brazier, F.M.T., Jonker, C.M. and Treur, J. (2000), Compositional design and reuse of a generic agent model, *Applied Artificial Intelligence*, **14**:491–538.

Brazier, F.M.T., Jonker, C.M., Treur, J. and Wijngaards, N.J.E. (2001), Compositional Design of a Generic Design Agent. *Design Studies journal*, **22**:439-471.

Brazier, F.M.T., Kubbe, O., Oskamp, A. and Wijngaards, N.J.E. (2002), Are Law-Abiding Agents Realistic? *in* Sartor, G. and Cevenini, C. (eds), *Proceedings of the workshop on the Law of Electronic Agents (LEA02)*, pp. 151-155.

Brazier, F.M.T., Oskamp, A., Prins, J.E.J., Schellekens, M.H.M. and Wijngaards, N.J.E. (2003). Are anonymous agents realistic? *in* Oskamp, A. and Weitzenböck, E. (eds), *Proceedings of the LEA 2003: The Law and Electronic Agents*, pp. 69-79.

Brazier, F.M.T., Oskamp, A., Schellekens, M.H.M. and Wijngaards, N.J.E. (2003a). Can Agents Close Contracts? *in* Oskamp, A. and Weitzenböck, E. (eds), *Proceedings of the LEA 2003: The Law and Electronic Agents*, pp. 9-20.

Brazier, F.M.T., Oskamp, A., Schellekens, M.H.M. and Wijngaards, N.J.E. (2003b). Are Mobile Agents Outlawed Processes? *in* Oskamp, A. and Weitzenböck, E. (eds), *Proceedings of the LEA 2003: The Law and Electronic Agents*, pp. 127-139.

Brazier, F.M.T., Overeinder, B.J., van Steen, M. and Wijngaards, N.J.E. (2002), Agent Factory: Generative Migration of Mobile Agents in Heterogeneous Environments, *in Proceedings of the 2002 ACM Symposium on Applied Computing (SAC 2002)*, pp. 101-106.

Bui, H.H., Kieronska, D. and Venkatesh, S. (1996), Learning other agents' preferences in multiagent negotiation, *in Proceedings of the National Conference on Artificial Intelligence (AAAI-96)*, pp. 114–119.

Burnett, R. (2001) International Carriage of Goods (Updates Chapter 2), in: Burnett, R. (2001), *Law of International Transactions.*

Castelfranchi, C (2000), Founding Agent's 'Autonomy' on dependence theory, *in Proceedings of ECAI 2000*, pp. 353-357.

Castelfranchi, C. (1998), Towards an agent ontology: Autonomy, delegation, adaptivity, *AI*IA Notizie*, **11**(3):45-50.

Castelfranchi, C. and Falcone, R. (1998), Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification, *in Third International Conference on Multi Agent Systems*, pp. 72-80.

Castelfranchi, C. and Falcone, R. (2000), Trust is Much More Than Subjective Probability: Mental Components and Sources of Trust, *in Proceedings 33rd Hawaii International Conference on System Sciences*. IEEE Press.

Cavanillas S. and Nadal A.M. (1999), *Research paper on contract law*, Electronic Commerce Legal Issues Platform (ECLIP) report, Deliverable 2.1.7bis.

Clift, J. (1999), Electronic Commerce: the UNCITRAL Model Law and Electronic Equivalents to Traditional Bills of Lading, *International Business Lawyer*, July/August, pp. 311-317.

Cooke, E. (2000), *The modern law of estoppel*, Oxford: Oxford University Press.

Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N. and Weerawarana, S. (2002), IEEE Internet Computing: Spotlight - Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI, *IEEE Distributed Systems Online*, **3**(4), http://dsonline.computer.org/0204/features/wp2spot.htm

Dale, J. and Mamdani, E. (2001), Open standards for interoperating agent-based systems. *Software Focus*, **2**(1):1–8, Spring.

Dastani, M., Jacobs, N., Jonker, C.M. and Treur, J. (2001), Modeling user preferences and mediating agents in electronic commerce, *in* F. Dignum and C. Sierra (eds), *Agent-Mediated Electronic Commerce*, Lecture Notes in AI, **1991**, pp. 164–196. Springer Verlag.

Dekker, K., Sycara, K. and Williamson, M. (1997), Middle-Agents for the Internet, *in Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence IJCAI '97*, pp. 578-583, Morgan Kaufmann.

Deutsch, M. (1962), Cooperation and Trust: Some Theoretical Notes, *in Nebraska Symposium on Motivation* Jones, M.R. (ed), pp. 275-318, Nebraska University Press.

Dijksterhuis-Wieten, H.L.G. (1998), Bewijsrecht in civiele procedures, Kluwer.

Ding, Y., Fensel, D., Klein, M. and Omelayenko, B. (2002), The Semantic Web: Yet Another Hip? *Data and Knowledge Engineering*, **41**(2/3):205-227.

Edwards, A. (1996), Bolero - A TTP Project for the Shipping Industry, *Information Security Technical Report*, **1**(1):40-45.

Erman, L. D. Hayes-Roth, F., Lesser, V.R. and Reddy, D.R. (1980), The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty, *ACM Computing Surveys*, **12**(2):213-253, reprinted in Webber and Nilsson (1981).

Esch, R.E. van (2002), Electronic commerce, *in* Prins, J.E.J. and Esch, R.E. van (eds), *Privacyregulering in theorie en praktijk*, Serie Recht en Praktijk, pp. 371-393, Deventer: Kluwer.

Faber, D. (1996), Electronic Bills of Lading, *Lloyd's maritime and commercial law quarterly*, **2**:232-244, May.

Falcone, R. and Castelfranchi, C. (2001), The Socio-cognitive Dynamics of Trust: Does Trust Create Trust? *in* Falcone, R., Singh, M.P. and Tan, Y-H. (eds), *Trust in Cyber-societies, Integrating the Human and Artificial Perspectives*, Lecture Notes in Computer Science, **2246**, pp. 55-72. Springer.

Falcone, R., Singh, M.P. and Tan, Y-H. (2001). Introduction: Bringing Together Humans and Artificial Agents in Cyber-Societies: A New Field of Trust Research, *in* Falcone, R., Singh, M.P. and Tan, Y-H. (eds), *Trust in Cyber-societies, Integrating the Human and Artificial Perspectives*, Lecture Notes in Computer Science, **2246**, pp. 1-8, Springer.

Finin, T., Labrou, Y. and Mayfield, J. (1997), KQML as an agent communication language, *in* J. Bradshaw (ed), *Software Agents*, pp. 291–316, MIT Press, Cambridge.

FIPA (2000), FIPA ACL message structure specification. http://www.fipa.org.

FIPA (2001), FIPA agent platform, 2001. http://www.fipa.org.

Fipa ACL (2002), *Fipa ACL Message Structure Specification*, SC00061G, Standard, 2002/12/03, and *Fipa Ontology Service Specification*, XC00086D, Experimental, 2001/08/10.

Franken, H. and Kaspersen, H.W.K. (2001), Strafrecht en strafvordering, *in* Franken, H., Kaspersen, H.W.K. and de Wild, A.H. (eds), *Recht en Computer*, in the series: Recht en Praktijk, Deventer: Kluwer.

Froomkin A.M. (1995), Anonymity and Its Enmities, *Journal Online Law*, art. 4, June.

Froomkin A.M. (1996), Flood Control on the Information Ocean: Living With Anonymity, Digital Cash, and Distributed Databases, *Pittsburgh Journal of Law and Commerce,* 395, 15 U.

Fuggetta, A., Picco, G.P. and Vigna, G. (1998), Understanding code mobility, *IEEE Transactions on Software Engineering*, **24**(5):342–361, May.

Fung, D. (1999), *Pre-contractual Rights and Remedies: Restitution and Promissory Estoppel*, Sweet & Maxwell Asia.

Gabber, E., Gibbons, P., Matias, Y. and Mayer, A. (1997), How to Make Personalized Web Browsing Simple, Secure, and Anonymous, *in* R. Hirschfeld (ed), *Financial Cryptography, Proceedings of the First International Conference, FC '97*, Anguilla, British West Indies, February 24-28, 1997. Springer-Verlag, LNCS, **1318**, pp 17-32.

Gambetta, D. (2000), Can We Trust Trust *in* Gambetta, D. (ed), *Trust: Making and Breaking Cooperative Relations*, pp. 213-237, Department of Sociology, University of Oxford.

Goldschlag, D.M., Reed, M.G. and Syverson, P.F. (1999), Onion Routing for Anonymous and Private Internet Connections, *Communications of the ACM*, **42**(2):39-41.

Grandison, T. and Sloman, M. (2000), A Survey of Trust in Internet Applications, *IEEE Communications Surveys*, **3**(4):2-16.

Gray, R.S., Cybenko, G., Kotz, D., Peterson, R.A. and Rus, D. (2002), D'Agents: Applications and performance of a mobile-agent system, *Software: Practice and Experience*, 32(6):543-573.

Grijpink, J.H.A.M. and Prins, J.E.J. (2001), New Rules for Anonymous Electronic Transactions? An Exploration of the Private Law Implications of Digital Anonymity, *The Journal of Information, Law and Technology (JILT)*, **2**.

Gruber, T.R. (1993), A translation approach to portable ontology specifications, *Knowledge Acquisition*, **5**(2):199-220.

Gudivada, V.N., Raghavan, V.V., Grosky, W.I. and Kasanagottu, R. (1997), Information retrieval on the World Wide Web. *IEEE Internet Computing*, **1**(5):58–68, September/October.

Hartkamp A.S. (2001), *Mr. C. Asser's handleiding tot de boefening van het Nederlands burgerlijk recht*. 4. Verbintenissenrecht. Deel II. Algemene leer der overeenkomsten, W.E.J. Tjeenk Willink.

Hartkamp A.S. and Tillema. M.M.M. (1995), *Contract law in the Netherlands*, Den Haag: Kluwer Law International 1995.

Hawkes, P. (1995), Supertag - Reading multiple devices in a field using a packet data communications protocol, i*n CardTech /Securtech '95*, April.

Hidma, T.R. and Rutgers, G.R. (1995), *Pitlo. Het Nederlands burgerlijk recht*. Deel 7. Bewijs, Gouda Quint.

Hofman-Ruigrok, M.M. (1994), Identificatieplicht, *PS* 1994, pp. 1536-1545.

Holland, J.H. (1995), *Hidden Order: How Adaptation Builds Complexity*. Perseus Books, Cambridge, Massachusetts.

Horst, R.J.M. van der (2002), De Wet Bescherming Persoonsgegevens, Gevolgen voor de organisatie en de automatisering, *in* R.E. van Esch en J.E.J. Prins (eds), *Privacyregulering in theorie en praktijk*, Serie recht en Praktijk nr. 75, Deventer, Kluwer.

IIDS group (2003), forthcoming technical report on security & Agentscape, Vrije Universiteit Amsterdam.

Jansen, W. (2001), A Privilege Management Scheme for Mobile Agent Systems, *in First International Workshop on Security of Mobile Multiagent Systems, Autonomous Agents Conference*, ACM Press.

Jennings, N.R. (2000), On agent-based software engineering. *Artificial Intelligence*, **117**(2):277–296, Mar.

Jennings, N.R. and Wooldridge, W.J. (1998), editors, *Agent Technology: Foundations, Application, and Markets*. Springer-Verlag, Berlin, Germany.

Jurca, R. and Faltings, B. (2002), Towards Incentive-Compatible Reputation Management, *in* Falcone, R., Barber, S., Korba, L. and Singh, M. (eds), *Proceedings of the Workshop 'Deception, Fraud and Trust' of the Autonomous Agents Conference*, pp. 92-100, ACM Press.

Karnik, N. and Tripathi, A. (2001), Security in the Ajanta Mobile Agent System, *Software – Practice & Experience*, **31**(4):301-329, Apr.

Kemna, A.M.Ch. (2001), De vraagstukken van bewijs en bewaring in een elektronische omgeving, *in* Kaspersen, H.W.K., Wild, A.H. de and Franken, H. (eds), *Recht en computer*, Kluwer.

Kendall, E. (1998) Agent Roles And Role Models: New Abstractions For Intelligent Agent System Analysis And Design, *in Proceedings of International Workshop on Intelligent Agents in Information and Process*

Management, Germany, September, 1998

Kephart, J.O. and Chess, D.M. (2003), The Vision of Autonomic Computing, *IEEE Computer*, **36**(1):41-50.

Labrou, Y., Finin, T. and Peng, Y. (1999), The current landscape of Agent Communication Languages, *IEEE Intelligent Systems*, **14**(2):45-52.

Lange, D.B. and Oshima, M. (1999), Seven Good Reasons for Mobile Agents, *Communications of the ACM*, March, 42(3):88-89.

Lange, D.B., Oshima, M., Karjoth, G. and Kosaka, K. (1996), Aglets: Programming mobile agents in Java, *in Worldwide Computing and Its Applications*, Lecture Notes in Computer Science, **1274**, pp. 253-266. Springer-Verlag, Berlin, Germany.

Laryea, E.T. (2001) Bolero Electronic Trade System – An Australian Perspective, *Journal of International Banking Law*, **16**(1):4-11.

Lawson, B. (1997), *How Designers Think: The Design Process Demystified*, 3rd edition. Oxford: Architectural Press.

Levy, A.Y., Sagiv, Y. and Srivastava, D. (1994), Towards efficient information gathering agents, *in Software Agents, Proceedings of the AAAI 1994 Spring Symposium*, pp. 64–70.

Lu, S. and Smolka, S.A. (1999), Model Checking the Secure Electronic Transaction Protocol, *in Proceedings of Seventh International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'99)*, pp. 358-365, ACM Press.

Maes, P. (1994), Agents that reduce work and information overload, *Communications of the ACM*, **37**(7):31–40, July.

Marin, O., Sens, P., Briot, J-P. and Guessoum, Z. (2001), Towards Adaptive Fault-Tolerance For Distributed Multi-Agent Systems, *in Proceedings of ERSADS'2001*, Bertinoro, Italy, pp. 195-201.

Marsh, S.P. (1994), *Formalising Trust as a Computational Concept. Department of Computing Science and Mathematics*, PhD thesis, University of Stirling.

Martin, D. (1998), Internet Anonymizing Techniques, in: *;login: Magazine,* May, http://www.usenix.org/ publications/login/1998-5/martin.html

Martin, D., Cheyer, A. and Moran, D. (1999), The open agent architecture: A framework for building distributed software systems, *Applied Artificial Intelligence*, **13**(1/2):91–128.

Matthijssen, L. (1999), *Interfacing between lawyers and computers*, An architecture for knowledge-based interfaces to legal databases (diss. Tilburg), Kluwer Law International, p. 29.

McWilliams, B. (2001), Stealing MS Passport's Wallet, *Wired News*, November 2, http://www.wired.com/news/technology/0,1282,48105,00.html

Meyer, J.-J.C. and Schobbens, P.-Y. (1999), *Formal Models of Agents, ESPRIT Project ModelAge Final Workshop, Selected Papers*, Lecture Notes in AI, **1760**, 253 pp, Springer Verlag.

Miller, M.L., Cox, I.J., Linnartz, J-P.M.G. and Kalker, T. (1999), A review of watermarking principles and practices, *in* K. K. Parhi and T. Nishitani (Eds), *Digital Signal Processing in Multimedia Systems*, chapter 18, pp. 461-485, Marcell Dekker Inc.

Milojicic, D., Breugst, M., Busse, I., Campbell, J., Covaci, S., Friedman, B., Kosaka, K., Lange, D., Ono, K., Oshima, M., Tham, C., Virdhagriswaran, S. and White, J. (1998), MASIF: The OMG mobile agent system interoperability facility, in *Proceedings of the 2nd International Workshop on Mobile Agents (MA'98)*, Lecture Notes in Computer Science, **1477**, pp. 50–67, Berlin, Germany, Sept. Springer-Verlag.

Mobach, D.G.A., Overeinder, B.J., Wijngaards, N.J.E. and Brazier, F.M.T. (2003), Managing Agent Life Cycles in Open Distributed Systems, *in Proceedings of the 18th ACM Symposium on Applied Computing*, pp. 61-65, ACM Press.

Mui, L., Mohtashemi, M. and Halberstadt, A. (2002), Notions of reputation in multi-agents systems: a review, *in Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pp. 280-287, ACM Press.

Nickerson, J.R., Chow, S.T. and Johnson, H.J. (2001), Tamper Resistant Software - Extending Trust into a Hostile Environment, *ACM Multimedia Workshop*, Ottawa, October 2001.

Nonaka, I. and Takeuchi, H. (1995), *The Knowledge-Creating Company*. Oxford University Press.

Noordende, G. van 't, Brazier, F.M.T. and Tanenbaum, A.S. (2002), A Security Framework for a Mobile Agent System, *in* K. Fischer and D. Hutter (eds), *Proceedings of the 2nd International Workshop on Security in Mobile Multiagent Systems (SEMAS 2002)*, associated with AAMAS-2002, Bologna, Italy , DFKI Research Report RR-02-03, Deutsches Forschungszentrum fur Künstliche Intelligenz, pp. 43-50.

NTT- DOCOMO *I-mode* http://www.nttdocomo.com/i/service/tokutyou.html

Nwana, H.S. (1996), Software agents: An overview, *The Knowledge Engineering Review*, **11**(3):205–244.

Odubiyi, J.B., Kocur, D.J., Weinstein, S.M., Wakim, N., Srivastava, S., Gokey, S. and Graham, J. (1997), SAIRE–A scalable agent-based information retrieval engine, *in Proceedings of the First International Conference on Autonomous Agents*, pp. 292-299, Marina del Rey, CA, Feb. 1997.

Omicini, A. and Papadopoulos, G.A. (2001), Coordination models and languages in AI, *Applied Artificial Intelligence*, **15**(1):1–103, Jan.

Oskamp, A. and Brazier, F.M.T. (2001), Intelligent agents for lawyers, *in Proceedings of the Workshop Legal Knowledge Systems in Action: Practical AI in Today's Law Offices*, 5 pages.

Patel-Schneider, P., Horrocks, I. and Harmelen, F. van (2002), Reviewing the Design of DAML+OIL: An Ontology Language for the Semantic Web, *in* R. Dechter, M. Kearns and R. Sutton (eds) *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pp. 792-797.

Peine, H. and Stolpmann, T. (1997), The architecture of the Ara platform for mobile agents, in *Proceedings of the First International Workshop on Mobile Agents (MA'97)*, Lecture Notes in Computer Science, **1219**, pp. 50-61, Berlin, Germany, Apr. Springer-Verlag.

Pfleeger P. (1997), *Security in Computing*, international edition Prentice Hall, pp. 158, ISBN 0-13-185794-0

Pham, Ph.N. (1994), The waning of promissory estoppel, Cornell Law Review, p. 1263, 1994.

Picco. G. P. (2001) Mobile agents: An introduction, *Microprocessors and Microsystems*, **25**(2):65–74, April.

Poslad, S and Calisti, M (2000), Towards Improved Trust and Security in FIPA Agent Platforms, *in Proceedings of the Autonomous Agents 2000 Workshop on Deception, Fraud and Trust in Agent Societies*, Barcelona, Spain, pp. 87-90.

Proctor, C. (1997), *The legal role of the bill of lading, sea waybill and multimodel transport document*, Pretoria: Interlegal.

Rao, A.S. and Georgeff, M.P. (1993), A model-theoretic approach to the verification of situated reasoning systems, *in Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, Chambéry, France, pp. 318-324.

Rao, A.S. and Georgeff, M.P. (1995), BDI agents: From theory to practice, *in Proceedings of the First Intl. Conference on Multiagent Systems*, pp. 312–319, San Francisco, CA.

Reiter, M.K. and Rubin, A.D. (1998), Crowds: Anonymity for Web Transactions, *ACM Transactions on Information and System Security,* **1**(1):66–92, November 1998.

Reiter, M.K. and Rubin, A.D. (1999), Anonymity Loves Company: Anonymous Web Transactions with Crowds, *Communications of the ACM*, **42**(2):32-38, February.

Rieke, A and Demuth, T. (2001), JANUS: Server Anonymity in the World Wide Web, *in* Gattiker, U.E. (ed), *Conference Proceedings EICAR International Conference*, pp. 195-208.

Roos, Th. de and Wissink, L. (1996), Uitingsdelicten op het Internet en strafrechtelijke repressie, *NJB* 1996/41, pp. 1728-1733.

Sander, T. and Tschudin, C.F. (1998), Protecting Mobile Agents Against Malicious Hosts, *in* Vigna, G. (ed), Mobile Agents and Security, Lecture Notes in Computer Science, **1419**, Springer-Verlag, pp. 44-60.

Sanders, R (2000), Goedkoop grossieren, *Computable*, 8 september 2000, **36**, p. 33, http://www.computable.nl/artikels/archief0/d36ag001.htm

Sandholm, T.W. (1999), Automated Negotiation, *Communications of the ACM*, **42**(3):84-85.

Schillo, M., Funk, P. and Rovatsos, M. (1999), Who can you trust: Dealing with Deception, *in* Falcone, R. (ed), *Proceedings of the Workshop 'Deception, Fraud and Trust' of the Autonomous Agents Conference*, ACM Press.

Serban. R. (2002), *The private cyberspace: Modeling Electronic Environments inhabited by privacy concerned Agents*, PhD thesis, Siks Dissertation Series No. 2002-5.

Shamir, A. (1979), How to share a secret, *Communications of the ACM*, **22**(11):612-613.

Sherif, M.H. (2000), *Protocols for Secure Electronic Commerce*, CRC Press, Advanced and Emerging Communications Technologies Series.

Shoham, Y. (1993), Agent-oriented programming, *Artificial Intelligence*, **60**(1):51–92, Mar.

Steen, S.J. van der (1988), *Identificatieplicht: identificatieplicht in Nederland en in diverse Westeuropeselanden alsmede de Verenigde Staten*, Ministerie van Binnenlandse Zaken.

Stephen, M. (2000), Print Your Next PC, *MIT Techreview*, November/December issue, http://www.techreview.com/magazine/nov00/mihm.asp

Suri, N., Bradshaw, J., Breedy, M.R., Groth, P.T., Hill, G.A. and Jeffers, R. (2000), Strong mobility and fine-grained resource control in NOMADS, *in Proceedings of the Joint Symposium on Agent Systems and Applications/Mobile Agents (ASA/MA2000)*, pp. 2–15, Zurich, Switzerland, September 2000.

Suri, N., Bradshaw, J.M., Breedy, M.R., Groth, P.T., Hill, G.A., Jeffers, R., Mitrovich, T.S., Pouliot, B.R., and Smith, D.S. (2000), Nomads: Toward a strong and safe mobile agent system, in *Proceedings of the Fourth International Conference on Autonomous Agents*, pp. 163–164.

Sycara, K. and Zeng, D. (1996), Multi-agent integration of information gathering and decision support, *in Proceedings of the 12th European Conference on Artificial Intelligence (ECAI'96)*, pp. 549–553.

Sycara, K., Paolucci, M., Velsen, M. van, and Giampapa, J. (2001), *The RETSINA MAS infrastructure*, Technical report CMU-RI-TR-01-05, Robotics Institute, Carnegie Mellon University, March.

Tanenbaum, A.S. and Van Steen, M. (2002), *Distributed Systems: Principles and Paradigms*, Prentice Hall, New Jersey.

Toorenburg, M.M. van (1998), *Medeplegen* (diss. Tilburg), Deventer: W.E.J. Tjeenk Willink.

Tripathi, A., Karnik. N., Vora, M., Ahmed, T. and Singh, R. (1999), Mobile agent programming in Ajanta, *in Proceedings of the 19th International Conference on Distributed Computing Systems (ICDCS'99)*, pp. 190-197, Austin, TX, May.

Villecco Bettelli, A. (2002), Agent technology and on-line data protection, *in* Sartor, G. (ed), *Proceedings of the workshop on the law and electronic agents, LEA2002*, pp. 53-57.

Want, R. and Russel D. M., (2000), *Ubiquitous Electronic Tagging*, Distributed Systems Online, **3**(4), http://computer.org/dsonline/archives/ds200/ds2wan.htm

Weitzenböck, E.M. (2001), *Electronic agents and the formation of contracts*, ECLIP report 2001.

Wellner, P., Mackay, W. and Gold, R. (1993), Computer augmented environments: Back to the real world, *Communications of the ACM*, **36**(7), August.

Wijngaards, N.J.E., Overeinder, B.J., van Steen, M. and Brazier, F.M.T. (2002), Supporting Internet-scale Multi-agent Systems, *Data and Knowledge Engineering*, **41**(2-3):229-245.

Williams, S. (1995), The PGP Web of Trust, *Byte*, February, http://www.byte.com/art/9502/sec13/art4

Willmott, S. N., Dale, J., Burg, B., Charlton, C. and O'brien, P. (2001), Agentcities: A Worldwide Open Agent Network, *Agentlink News*, **8**:13-15, November, http://www.AgentLink.org/newsletter/8/AL-8.pdf

Winslett, M, Yu, T., Seamons, K.E., Hess, A., Jacobson, J. Jarvis, R., Smith, B. and Yu, L. (2002), Negotiating Trust on the Web, *IEEE Internet Computing, 6(6)*:30-37.

Witkowski, M., Artikis, A. and Pitt, J. (2001), Experiments in Building Experiential Trust in a Society of Objective-Trust Based Agents, *in* Falcone, R., Singh, M.P. and Tan, Y-H. (eds), *Trust in Cyber-societies,*

Integrating the Human and Artificial Perspectives, Lecture Notes in Computer Science, **2246**, pp. 111-132. Springer.

Wooldridge, M.J. and Jennings, N.R. (1995), Intelligent Agents: Theory and practice. *The Knowledge Engineering Review*, **10**(2):115–152.

Wooldridge, M.J. and Jennings, N.R. (1999), Software Engineering with Agents: Pitfalls and pratfalls. *IEEE Internet Computing*, **3**(3):20–27.

Zhou, J. and Gollmann, D. (1996), A fair non-repudiation protocol, *in Proceedings of the 15th IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, pp. 55-61.

Zwalve, W.J. (2000), *C.Æ. Uniken Venema's Common Law & Civil Law*, Deventer: W.E.J. Tjeenk Willink.

A SCENARIOS

A.1 INTRODUCTION

Four different scenarios in which agents play different roles are described below. These descriptions are not meant to be complete, nor necessarily feasible: they are meant to identify the types of tasks agents may possibly perform in different areas of application. In combination these four scenarios cover a broad and diverse area of agent applications: grocery shopping, chemical commodities marketplace, hospital, and local government.

A.1.1 Grocery Shopping

In this scenario, customers use shopping carts containing intelligent agents to shop in a grocery store; the intelligent shopping cart keeps track of the goods to be bought, as well as provide information about products and advertisements. This scenario involves a business (the grocery shop), and customers, whereby the intelligent agents are provided by the grocery shop: a closed environment.

In this scenario, intermediary concepts such as identity, anonymity and contract closing play an important role.

A.1.2 Chemical Commodities Market

In this scenario, buyers and sellers of chemical substances meet with the intended goal to buy and sell chemical goods. Agents negotiate and transact on behalf of their buyers and/or sellers. This is modelled as an open environment, whereby any agent can enter the marketplace (provided it adheres to some protocols and possibly represents a known, human, buyer/seller). Businesses are mainly involved in this marketplace: a B2B system.

In this scenario, intermediary concepts such as autonomy, contract closing, identity and anonymity, and integrity play an important role.

A.1.3 Hospital

In this secnario, information management within a hospital is facilitated by agents which manage patient files, work schedules, etc. These agents are able to provide humans with the information needed, while preventing unncessary information provision. Agents representing third-parties such as insurance companies, may also require information from patient files: protocols including authentication and authorisation play a major role. Agents accessing this system are restricted to known agents from known parties: the hospital needs to be a closed environment, stressing privacy and confidentiality. This is mostly an internal system supporting hospital staff; while third parties interact with the hospital in a business-to-business setting.

In this scenario, intermediary concepts such as identity, integrity, integrity of evidentiary data,, confidentiality, reliability and non-excessiveness play an important role.

A.1.4 Local Government

In this scenario, business processes in local government are supported by agents, including planning and information provision. Information maintained by the agents concerns citizens, governmental regulations, local regulations, etc. Adequate provision of relevant information to an employee, on time, involves a good understanding of prevalent business processes. This system is an in-company system: a closed environment.

In this scenario, intermediary concepts such as identity, integrity, confidentiality, reliability, and non-excessiveness play an important role.

A.1.5 Open & Closed Systems

Analysis of the characteristics of the four scenarios resulted in (1) a cluster of scenarios in which information access is not inherently restricted, and (2) a cluster of scenarios in which information access is necessarily restricted. Within these clusters information processing itself may be comparable.

A.1.5.1 Open systems

The chemical marketplace and grocery shopping scenarios are primarily open systems, and can be compared to open markets.

These scenarios can be characterised by the following processes:

1. Provision of information
    - Individually
    - Blackboard
    - Intermediary
2. Processing of information: Negotiation
    - Direct negotiation
    - Auction
    - Matchmaking
3. Processing of information: Protocol in de broadest sense of the word, including finalisation of deals.
4. Payment and transactions
5. Transport

A.1.5.1 Closed Systems

The hospital and local government scenarios are primarily closed systems: scenarios in which information access within an organisation is necessarily restricted. These scenarios can be characterised by the following processes:

1. Access
2. Analysis of data
3. Processing of data
4. Planning/resource management

A.2 GROCERY SHOPPING

A.2.1 Context

In their article Want and Russel (2000) predict that all physical goods that are for sale will, in the not too distant future, be labelled in such a way that they are electronically accessible. A number of large companies have indicated that, should it be possible to build such electronic devices for less than a penny, they will be used for new marketing possibilities. Several technological advances seem to indicate that this prediction indeed will exist in the not-too-far-future. Plastic electronics are possible and also new techniques are being developed to print electronic circuits (Stephen, 2000). At the same time there are developments that make chips so efficient that they can be activated with a low power inductive power source so that the chip can generate a signal with a data content equal to or even more of a bar code. In Japan cellular phones are used that give the user the possibility to do micro payments based on the amount of data sent (NTT Docomo).

In the fifties several packaging methods were developed that are devised in such a way that they only can be opened once. It will not be difficult to envision packages that when opened will change the status of the packaging. In the not too far future these possibilities with an already existing appropriate ‘proven’ supporting infrastructure will be available. What is the use?

Nowadays so called Internet-ready household appliances are made. A refrigerator with a tagging system would know when the expiration date of a milk carton is, even knowing that the expiration date is shortened because the carton has been opened. A supermarket could on the basis of these technologies conduct its logistics on a real time basis: the moment a customer grabs a milk carton it receives another status. It is easy to see that this technology has social repercussions; e.g., a cash register is not needed anymore and the associated job of cashier becomes obsolete. Taking an object from a shelf could in fact mean the immediate buying of the object, when permitted.

With the infrastructure of micro payments it becomes possible that the customer in the context of the store can actually sell the object when he has not damaged the packaging and places it on the shelf. The role of a person must than be clear in the context of the store. Otherwise it is not possible to correctly handle an accident caused by a shop assistant. In some cultures this will lead to worse working conditions, because the responsibilities can be assigned meticulously. When there is no good model of the human behaviour patters with regard to our spending, this could also lead to a diminishing diversity of supply. In this environment agents could come in handy. An agent could detect for example that a packaging is damaged. A micro payment could be assisted by an

agent, an privacy agent could enforce a policy of non-individualization of the goods you are buying. Your refrigerator could tell your personal agent that you need to buy some milk for your child.

A.2.2 Agents

In the grocery trace the shopping cart agent plays a central role. The agent operates from a computer module that is attached to a run-of-the-mill shopping cart as used in your average supermarket. The shopping cart agent is meant to perform several functions:

- It facilitates the determination of what products a customer has put in his shopping cart, the calculation of the amount he has to pay and initiates payment. The cumbersome handling of goods at the till in supermarkets can thus be enormously alleviated.
- Protection of the property of the supermarket: to a certain extent the shopping cart agent complicates shoplifting, enlarges the chance of discovery and diminishes the chance that unpaid goods leave the supermarket by accident.
- The shopping cart agent can be used as a means to communicate special offers and other information that might be relevant from the shopkeeper to the customer.
- The shopping cart agent facilitates the navigation of the customer through the supermarket.

A.2.3 Example

This scenario describes a number of roles agents could play in grocery shopping, assuming a shopping list is available (either in machine-readable form or in a human being's mind). Two situations are described: one in which a human being determines the groceries to be bought, and the other in which a PDA is aware of the groceries to be acquired.

A.2.3.1 Situation 1: human grocery shopping

In this scenario a human being visits the supermarket. The supermarket uses shopping carts with electronic agents to interact with the customer. Both the products in a supermarket and the shopping cart are equipped with devices to register necessary information and interaction. Products all have (electronic) tags which can be detected by shopping cart agents. (Hawkes, 1995; Want and Russel, 2000). Each shopping cart keeps track of the total value of its contents and is equipped to facilitate electronic payment.

In addition the shopping cart agent can, for example, provide a customer with additional information. For example, it could inform him or her that a specific good in the vicinity of the cart is on special offer this week (Wellner, Mackay and Gold, 1993). A shopping cart agent can also automatically display information on "related" products. How "related" is defined depends on the types of information that may be valued. Related can be defined as comparable to the arrangement of products on shelves (e.g., milk is stored in the vicinity of yoghurt, red wine is next to the white wine), but can also be based on completely different criteria (e.g., when choosing red meat information on appropriate red wine is provided).

Apart from customer interaction, the product tags can also be used for other purposes. First of all, for pricing and positioning information. Every item sold can be tracked almost real-time. This makes it possible to further refine the logistics of the supermarket. The manufacturer of the product can store information in the tags such as amount, the caloric value, chemical composition, storage life, and other information such as a recipe or method of preparation. This factory information can make a refrigerator interactive: when a pack of milk is stored in the refrigerator a refrigerator-agent may draw knowledge from the packaging's tag about the expiration date, even that it is shortened by the opening of the package.

A.2.3.2 Situation 2: shopping with personal assistant

In this situation a personal assistant is introduced in the supermarket environment. This personal assistant can come in the form of a Personal Digital Assistant (PDA) or Personal Wearable Assistant (PWA). This personal assistant can interact with different types of entities. First, and most importantly, interaction with its owner: the personal assistant can make shopping lists on the basis of its knowledge of the user.

At home, agents such as a refrigerator agent can detect that milk for the growing baby is needed and adds this to the personal assistant's shopping list. The vacuum cleaner could indicate that it needs a new filter. The cat's litter box could indicate that it needs refilling.

In the supermarket, shopping cart agents can use a shopping profile provided by the personal assistant, make special offers that are of interest to the customer based on his or her profile known to the personal assistant. From a data protection point view this is interesting because this profile can be kept anonymous. The supermarket is not prohibited in its ability to use data mining techniques to operate more efficiently, but cannot link this data to individuals. For the customer there is no incentive to produce misguided information (doing so would lead to offers that are of less interest).

A.2.3.3 Issues with grocery shopping

One of the issues involved in this scenario is the increased digitisation of all kinds of physical objects, enabling agents to trace and detect physical objects. These 'tags' may be subject to viruses or other (physical) hacking attempts. It is easy, for example, to envision a scenario whereby agents try to fool another system or are fooled into believing that a bar of chocolate has the price of X plus 1 euro-cent, constituting a salami-attack (Pfleeger, 1997). In addition, these tags can be used to derive meta knowledge about a specific individual when combined with other sources, thus creating data protection issues.

Another issue is the correctness of the system, in a survey by a daily newspaper it was found that the bills where categorically incorrect, producing in overall errors that mostly where beneficial to the grocery store (although in some cases a customer was the beneficiary). Most customers do not bother because the amount of money involved is too little to warrant the effort, however this still is stealing. How is this resolved? What kinds of problems are associated with agent reconfiguration? For example: has the payment agent correctly been reconfigured to operate the money machine. Is the money machine activated in the correct context? For example: does the human see the right price or has the price been switched with another human? The collaboration of the agents could also lead to errors. When the personal assistant releases the wrong profile, (for example a profile for a catering shop where its customer works or a different supermarket) this could lead to unexpected behaviour in the interaction. Who is responsible? The agent builder? The user who was not adequate in stating its role? The supermarket not being able to tell who he/she was?

A.2 CHEMICAL COMMODITIES MARKET

In this scenario the hypothetical marketplace Chemicality.com is presented. Chemicality.com is an electronic, on-line marketplace, which is supported by agents. The marketplace offers a platform through which buyers and sellers can trade in basic chemical commodities like caustic soda, solvents and acids. Chemicality.com uses a number of ways to facilitate trading in chemical commodities. First of all they offer a so called blackboard market place. Such a blackboard can be compared to a board with advertisements hanging in a supermarket. So agents can make advertisements according to a format provided by Chemicality.com. These advertisements are published on the website of Chemicality.com and are stored in a database of Chemicality.com. In case another agent is sufficiently interested to react to an advertisement, the agent sends a message to both the original agent and Chemicality.com stating its intersst in advertised commodities.

Chemicality.com also offers the possibility of an on-line auction. In this auction several agents are found that offer chemical commodities. In contrast to the blackboard, in this situation it is possible for more then one agent to place bids during a specific amount of time. The agent with the highest bid receives the commodities offered by another agent. Chemicality.com co-ordinates the auction and retains auction information, using a website and a database.

In all situations, Chemicality.com uses a format including amongst others the names, (cryptographic) protocols, procedures, authorizations/permissions, access control and logs. These elements are necessary and compulsory in order to identify and trace agents involved.

A.2.1 Context

In the chemical commodities market there are no central marketplaces. Vendors of supplies cannot be found easily or have busy agendas and are difficult to contact. Using the Internet, electronic, on-line marketplaces can be constructed. (Sanders, 2000) An electronic marketplace offers at least a central communication channel through which supply and demand can be matched. This matching can take place in several forms: a 'blackboard' (Erman, Hayes-Roth, Lesser and Reddy, 1980), whereby interested parties match their needs based on advertisements on a publicly available meeting place; a marketplace with (automated) matchmaking through a third party; an auction; and finally through (automated) negotiation.

A.2.2 Agents

Chemicality.com is an electronic, on-line marketplace, which is supported by agents. The marketplace offers a platform through which buyers and sellers can trade in basic chemical commodities like caustic soda, solvents and acids. Chemicality.com uses a number of ways to facilitate trading in chemical commodities. First, all variables (such as grade, concentration, specs, delivery details) concerning the commodities are standardised. Secondly, both buyers and sellers are screened before they are allowed access to the marketplace. To ensure payment of the seller, credit insurance is offered. Thirdly, all communication is encrypted.

A.2.3 Examples of Chemical Marketplaces

Five alternative situations for a chemical marketplace:

- 'blackboard' marketplace
- auction hall
- matchmaking
- automated negotiation
- transportation

A.2.3.1 Situation1: The 'blackboard' marketplace

In a blackboard based marketplace both buyer and seller agents post messages on a blackboard. This blackboard is, in general, visible to all parties in the market place. Buyer agents can post messages asking for specific goods possibly including detailed information on price ranges, quantities, etc. Seller agents place offers on the blackboard, again possibly including more detailed information. Negotiation can take place directly between the parties involved. Whether the identity of the parties is visible on the blackboard depends on the specific instantiation.

A.2.3.2 Situation 2: The Auction hall

An auction hall with an auctioneer agent is provided with goods by seller agents. In this situation the auctioneer agent uses a Dutch auction to sell the chemicals to the buyer agents. The mechanism of a Dutch auction works like this: a chemical is up for auction with a maximum price in 'front' of the auction hall filled with buyers (agents). The prices start to drop with a predetermined rate and time interval. A buyer agents who represents the company does a bid when the price is to its liking. The agent that first does the bid acquires the goods, and only has to tell the auctioneer how much it needs. When agents participate in an auction it is possible that collectives of buyer agents are formed to bargain for lower prices or for reseller-agents to match the demand of a customer.

A.2.3.3 Situation 3: Match making

In this situation Chemicality.com offers a matchmaking service via a middleman, in fact a directory service. The middleman generates a list of possible suppliers or buyers depending on the role of the customer agent, on request. Buyer and seller agents directly negotiate with each other. This matchmaking agent can be extended to form groups of buyers and sellers or to match buyers and sellers based on knowledge of dependability of the parties, their credit-rating, nationality, the kind of business they run or other relevant parameters. If needed the anonymity for the 'pre-negotiating' parties can also be guaranteed by entrusting the matchmaking agent as a trusted-third-party.

A.2.3.4 Situation 4: Automated negotiation

In a more futuristic setting, agent technology can be used to automate the negotiations between buyers and sellers after the matchmaking agent has made the pre-selection, or after agents have noticed interesting offers/bids on a blackboard. Both buyer and seller agents would need to know the necessary constraints: a buyer agent would need to be able to make sure the product is delivered before a certain date. A seller agent could probably be able to guarantee delivery of goods within a certain period, but would require higher financial compensation. These agents are sent to the marketplace and are fully authorised to negotiate and finalise the deal.

A.2.3.5 Situation 5: Transportation

The next stage can be to automate transport of goods. Instead of a one to one relation between buyer and seller, a third agent is introduced to tender for the transport of the chemicals. The negotiating parties can decide to do a joint tender, but it is also possible that one of the parties does a tender because it carries the cost of transport.

A.2.3.6 Issues with agents in the marketplace

In situation 2 the auction hall makes use of an auctioneer agent. How is it guaranteed that the agent is not in collusion with itself or other parties? For example: with a sealed bid second price auction, also known as a Vickrey auction, the highest bidder wins, and the clearing price, the price that the winner has to pay, is equal to the second highest bid. It is possible that the auctioneering agent in future auctions takes the current winning bid as the reservation price, taking advantage of the fact that it can collect information about past bids of the bidders.

In situation 3 the marketplace can construct the anonymity of the parties, in the matchmaking procedure there are probably no problems to be expected. However, in what way is this anonymity protected, when is the anonymity revoked and how? This becomes even more pressing when the agents in the subsequent situations can operate anonymously. For example, when both parties negotiate anonymously, what happens if one party cheats on the other? Who is responsible, are (marketplace) policies needed?

Another concern involves the implementation of the negotiation algorithms; e.g., when a user of a negotiation agent is not satisfied with the performance of its agent, or suspects that agent does not use the negotiation algorithms according to the 'advertised' specifications, who will be held responsible? The programmer who implemented the software? The scientist who worked out the negotiation strategy? The company building the agent or the marketplace which leases the agents to customers?

How the marketplace is constructed can be of importance, some marketplaces could work in turns giving all agents an equal opportunity to take part in the marketplace. However it is also possible that the marketplace operates continuously. When there are agents that can travel to different marketplaces these agents could have problems adapting to the different marketplaces. How should this be resolved? A solution could be that every marketplace provides libraries to their 'customer' agents. But this may lead to unexpected behaviour in the agent. How is this resolved? When the marketplace is not available this can disrupt internal processes of the participating chemical companies, because they cannot sell or buy chemicals. Who is responsible? When collectives of agents are formed and they autonomously buy chemicals, what happens if one (or more) of the agents cannot fulfil its obligation?

A.3 THE HOSPITAL

A.3.1 Context

Hospital have major scheduling problems: patients, nurses, doctors, visiting hours, operating rooms, medical machine operations (x-ray, ECG, etc), administering medicine: all must be planned. Agents can be used to plan dynamically in this environment. It is also a heterogeneous environment, for nearly every task medical information needs to be available regarding a patient, which are needed by information agents.

Information in such an environment is sensitive, privacy (the other term is data-protection) aspects come into play. A nurse who is off duty or works in different departments, should not be concerned with a patient for whom he or she, at that moment, is not responsible. So an information guard agent is also needed to not only authorise, but also scrutinise the requests from information agents, doctors, etc. There are situations where particular data can be anonymised (for example, some research programs don't need specifics on persons), but also cost analysis to improve the efficiency of the hospital can use less detailed information. So far only the internal matters with a strict medical nature of the hospital are discussed.

Often several companies are active within a hospital, to provide services for patients; e.g., telephone and television services to hospital beds. Also the food catering is a mixed enterprise with some patients needing special food requirements and others standard food requirements. The outside world also has many and diverse interactions with a hospital. A medical helicopter or an ambulance with first aid teams can benefit from information agents regarding a possible medical history stored at some hospital or family doctor. In addition, insurance companies often need specific information regarding authorized prescriptions, costs of being in the hospital, etc.

This scenario describes the Intensive Care (IC) department of an academic hospital in Amsterdam. The purpose of the IC department is to guard and stabilise patients whose conditions are threatening. To reach this goal, the

department is supported by medical specialists, nurses, a secretary and medical equipment (e.g., medicines or machines).

When a patient is in a life-threatening condition, then he or she is moved to the IC department. At this department the patient's condition is diagnosed and a treatment for guarding and stabilising the patient is given. Therefore the nurses and specialists are equipped with beepers and the systems are equipped with alarm systems. These systems for example, start to function when a machine diagnoses a blood pressure which is too low and by using sound signals the nurses and specialists are warned.

The IC department can be seen as an independent and centralised specialism, which takes care of patients in need of *intensive care*. If necessary, the assistance of specialists of other departments is called for. The responsibility for the treatment of IC patients lies in the hands of the IC department. After a patient has been diagnosed as stable, he or she, if necessary, is moved to another department of the hospital.

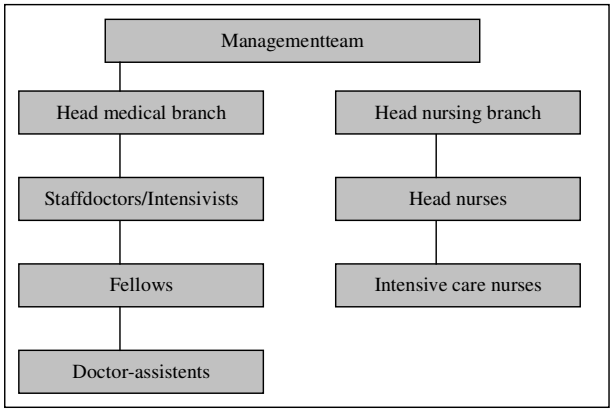


Figure A.1. Overview of organisation of the example IC department.

A.3.1.1 The medical division

The medical division is managed by a head of this division, who is also a member of the management team. This head is involved in management tasks like determine budgets and directing the department. Direct under the head of the medical branch are the staffdoctors/intensivists. Their tasks involve patient care, performing research and giving and receiving education. The fellows are the employees of the department who are trained to be a intensivist like longarts or cardiologist. They are also involved in patient care, performing research and giving and receiving education. The doctor-assistants are among others PhD. students and are involved in patient care and receiving education. Between the employees of the medical division periodic consultations take place.

A.3.1.2 The nursing division

The nursing division is guided by a head-nurse. This head is responsible for management and education of this division. The head of nurses reports directly to the head of nursing. The head of nurses is involved with management and patient care. The intensive care nurses are involved with patient care and receiving education. Between the employees of the nursing division periodic consultations take place.

There are also periodic consultations between both divisions and between the management team and the management of the divisions.

A.3.1.3 Knowledge management

Patient data like daily reports is stored in files. These files are not electronically available and are kept at the department as long as the patient is admitted. When a patient is released from the hospital, his or her files are stored in the archive of the hospital. When, for example, another hospital needs a patients file, then this file is physically moved by a courier. To gain knowledge, the employees study files and use periodic consultations (for example in the form of referees by doctors and the discussion of themes by nurses). The employees also use (mandatory) courses and information which is available through the internet. This information consists for example of tasks performed at other IC departments in the Netherlands or results of medical research (for instance performed by the academic hospital itself) that can be used by doctors.

A.3.1.4 Information Technology (IT) support

The IC department is supported by different types of IT. IT is used for administrative tasks, but also for guarding and stabilising patients. Next a global list follows of used IT within the IC department.

The administration

The secretary of the IC department uses text editors to write letters. Nurses and doctors use Internet and e-mail. By using the spreadsheet application Microsoft Excel, databases are maintained. These databases contain for instance data about the different kinds of education and data about meetings like who, where and when.

Patient guarding

The IC uses the following equipment to support the guarding and stabilisation of patients:

- Monitors
  Monitors are used to provide information about for instance the heartbeat, the blood pressure and other pressures. This information can be printed on screen or on paper.
- Mechanical ventilator
- This machine takes care of the ventilation of the patient and shows volume and pressure curves.
- Pumps
  With the use of pumps the injection of fluids and medication is controlled.
- Balloon pump
  The balloon pump supports the heart.
- Dialysis equipment
  Dialysis equipment is used for artificial kidney treatment.
- Electrical beds
  Certain beds have the possibility to automatically adjust the height of the bed and to calculate the weight of the patient.
- Defibrillator
  This machine is used for the electrical treatment of heart-rhythm disorders.
- Blood gas machinery
  This machine analyses the adequacy of a patient's breathing.
- Electrocardiography (ECG)
  With support of an ECG, electrical signals of the heart are monitored.

The analyses performed by the different machines are, if necessary, printed and added to a file.

A.3.1.5 Problems

Within the IC department several problems exist.

A fast exchange of data is not possible because the files are not digitally available. When for example, another hospital wants to have a file, a courier is called to transport the file. This may cause a delay. Also the exchange of (x-ray) photo's is not yet digitally controlled. In the current situation could occur that a photo taken in the morning at another department arrives in the afternoon because the distribution took a lot of time.

The planning of nurses and doctors is (for the greater part) done manually. The data gained for planning is manually transferred to the personnel department. This department enters the data in to their information system. Updating, controlling and distributing the planning is a very labour intensive process.

The handwriting of doctors is often hard to read. When a doctor, for example, writes a prescription which can't be read by the nurses, the prescription is sent back to the doctor. The doctor has to rewrite the prescription, this time in a legible form. This process can take a long time.

A.3.1.6 Wishes

- A quick processing of planning.
- Digitally processing and availability of data.
- The security of the IC department is an important issue.
- Automatically connecting results of patient data.
- Automatically monitoring the stock of medicines and use of equipment.
- Automatically alarming and adjusting equipment when changes in the patient's situation occur.

A.3.1.7 Solutions

1. Agents could be used to plan dynamically in this environment.
2. Information agents

(to be expanded)

A.3.1.8 Current research

In the USA there are several patient information systems used to deal with these issues, to my knowledge not with agents. The Norwegian Professor G.Hartvigsen at the University of Tromsø has a research project that is called the DIPATO project . The project aims to integrate electronic patient record (EPR) data through the national computer network for health institutions ("Helsenett"). The foundation for the DIPATO project is the ongoing co-operation between the Norwegian Centre of Telemedicine, University Hospital of Tromsø, the Norwegian Centre for Medical Informatics (KITH) and the Department of Computer Science, University of Tromsø, on mobile agent technology and security, the Department of Community Medicine, University of Tromsø, on medical and epidemiological issues, and the Telenor Research, Tromsø, on EPR systems. In addition, the Department of Philosophy, University of Tromsø, addresses ethical problems.

In a slightly different but related context the Norwegian Professor G.Hartvigsen (REF) at the University of Tromsø also has a research project that is called "The Good Room" project (GRO). The project focuses on home-based caring service. The overall goal is to create a complete service package for senior citizens at their own homes. The project aims to establish an adequate infrastructure in private homes, at home-based caring service centrals and to private and public services and institutions. The Good Room project is organized as an open and including project under the "Campus Tromsø"-umbrella. Sub goals are:

1. Guaranteed data deliverance (fault tolerance). All alarm signals must be delivered at the home-based caring service central responsible for the area where the alarm was activated.
2. Security and privacy. The system shall protect confidential patient data from any form of abuse.
3. The project shall offer a complete service package for senior citizen
4. Cost effectiveness (compared to institutionalisation)

Other important issues include:

1. Physical consequences
2. Psychological consequences
3. Consequences for the community

A.3.2 Agents

In this scenario the basic assumption is that agents are directly or indirectly related to some goal that is important to humans. In the hospital, humans perform complex decisions and tasks. It seems in this situation not likely for the foreseeable future to assume that agents can, for example, replace the doctor or nurse. In this situation it is useful to distinguish between occupations played by humans and agents to emphasize the constructed hierarchical relationships that exist between humans and agents. To this end the terms ‘persona’ and ‘role’ are introduced. The term ‘persona’ describes 'The entities system of adaptation to, or the manner it assumes in dealing with, the world’. The word comes from Latin and it means "actor's mask.". Because a person can have several occupations the term ‘persona’ allows for compositionality: the person is a private man, has an occupation with another organization, etc., etc. In the hospital domain specific persona’s can be identified: a doctor, a nurse, a manager, etc. The term role is reserved for software agents (Kendall, 1999).

In the hospital, three domains can be identified in which the collaborative nature of the hospital can be improved. These are collaborative planning, dossier-management, and data distribution. Collaborative planning because the persona’s collaborate in a highly dynamic planning environment. Dossier management, because the focus of attention is the patient. Dossier management means storing all the task related data, needed to improve the condition of the patient, and data distribution to improve the persona’s functioning. For example, the nurse needs to transport the patient to the MRI-scanner; he has to know about the condition of the patient, he has to know which MRI-scanner to use, which doctor to contact, etc. All of these domains have in common that they perform secondary supportive functions that are nevertheless essential to the workflow of persona. So, three supportive roles for agents are identified.

- A persona’s planning agent
- A persona’s dossier management agent
- A persona’s knowledge management agent.

The persona’s planning agent has several roles to play:

1. The agent has a negotiator role with other persona's agents. Its task is to optimise the planning.
2. The agent has a gatherer role: its task is to gather planning data in order to be able to negotiate with other planning agents.
3. The agent can have an advisory role to its persona: an advisory task because the person can have other wishes.
4. The agent has a slave-master role in relation to the persona. Its task is to serve the persona.

A persona's dossier management agent has its own roles:

1. The agent has a gatherer of data role. This agent has the primary task to gather all data related to his persona (patient).
2. The agent is a protector of data, it has the task to enforce policies that are set by its persona. These policies are for the most part identified by the hierarchies of persona's working within the hospital. So a doctor has almost complete access to the dossier. Nurses usually only have to access treatment information, billing only has access to documents signed by doctors authorizing some course of treatment.
3. The agent has a slave-master role in relation to the persona. Its task is to serve the persona.

The persona's knowledge management agent:

1. This agent has the role of gatherer of data mostly in an event-based manner. Its task is to gather information relevant to the persona.
2. This agent has the role of distributor of data. Its task is to distribute data specific to the persona's it has relations with. These relationships usually have temporary, location and event based properties. For example: a nurse off duty is not likely to derive any meaningful information when he is lying on the beach. However when the nurse is on duty the nurse can receive signals from equipment that support the patient.
3. The agent has a slave-master role in relation to the persona. Its task is to serve the persona.

A.4 LOCAL GOVERNMENT

A.4.1 Context

Within government offices many parties interact, a dynamic environment in which agents may play a role in information provision as part of a long-term strategy to automate back-offices. The relevant business processes involve procedures, workflows, etc. that have to be prioritised and resource balanced.

For example, if a citizen wants a new passport this process has a high priority for the agency (earning money is an important objective), involving people to handle the procedure, a sufficient supply of 'blank' passports, etc. The city of Amsterdam, e.g., has systems that describe these procedures but does not have the means to couple these systems with everyday practice. When a substantial amount of the workforce has the flu, an agent could help in the planning, rescheduling of people and knowledge distribution with regard to the workforce. In this respect knowledge management is also an important requirement with important data protection issues attached, comparable with the aforementioned hospital case.

A.4.2 Agents

In this scenario, the management of resources (humans and knowledge) is discussed within the context of civil government interacting with its citizens. The civil governments have an ever-increasing body of diverse knowledge that has to be maintained to organise the environment of the citizens. Issues are town regulations that are unique within each town, the procedures to acquire a passport, the registration of newly born children, decease of family, permits, taxes, housing, health, waste disposal and a host of other diverse rules and recommendations. This scenario discusses the use of agents in the relation between civil government and citizens. An assumption made in this scenario is that the citizen's goals can be expressed in a machine-readable environment.

A.4.3 Example

A.4.3.1 Situation 1: Just in time information management

Some citizens have to visit the city hall to get -for example- their driver license. Citizens also phone the town hall to do their enquiries, while other search on the web their information. Counter personnel handles these requests and have thus three channels of interaction with a citizen: face-to-face, communication via the phone

and via the Internet. In a sense, the counter environment is a 'production' facility having such a knowledge intensive character that it requires automation. In the following paragraph the so-called 'counter assistant agent' is introduced to aid in the requests that citizens have.

Counter assistant agent

To assist the counter personal 'counter assistant' agents are needed that search for the required information based on the questions of the customer. In this situation, the procedure for the hand out of passports will be used to show the complex interactions that are possible (Augier, Shariq and Vedelo, 2001). When a citizen wants a passport, several prerequisites have to be fulfilled in order to commence the production of the passport. In al three channels of interaction, previously described, this agent can assist. When an information request is coming via the Internet, the agent can independently disseminate information to the customer and collect required information.

However, when the customer uses the phone, the agent is slightly less autonomous and instead, assists the counter person by showing the procedure that should be followed on its monitor. In the face-to-face situation, the counter assistant agent can place the human in the right context of the procedure. A context could be, for example, the citizen has via the Internet requested for the passport and the counter assistant agent has started the procedure for the production of the passport. Now when the customer is coming in to get his passport, the counter assistant agent can recognise this situation and places the counter person on duty in the right phase of the procedure. The counter assistant agent has thus the ability to recognise the contexts in which a procedure can take place and can proactively act on this.

Furthermore, the counter assistant agent detects trends in the customer behaviour. When certain questions are often raised by citizens face-to-face, the counter assistant agent could detect that the answer cannot be found online, and creates a faq, which after review, is published. In this situation the agents *add* to the body of knowledge.

The advantages of such an agent are fairly obvious: there is less a priori knowledge needed by the human counter person to assist the citizen and thus should be able to assist customers on more different fields of expertise.

A.4.3.2 Situation 2: Labour resource assistance

The first situation describes an agent that could assist with procedures. However, often procedures have different priorities within an organisation. The previously discussed passport procedure is important to citizens and an important source of income for the town. This procedure has a higher priority in comparison to, for example, the request for a permit to build a shed dormer.

In addition, the availability of human labour is an issue. Labour resource management to prioritise the human labour is then an issue.

Labour resource assistant agent

A 'labour resource assistant agent' is needed. To do this each employee is represented by his or her own personal assistant agent. The labour resource assistant agent keeps track of the personal assistant agents and prioritises work on basis of existing policies (for a discussion on distributed scheduling see also Brazier, Jonker, Jüngen, and Treur, 1999). The labour resource assistant agent can request that certain people who are working at home should instead come to work at the counter.

The agent could also call an employment agency to hire more people. In the case that a lot of people have called in sick, the agent can start a procedure to inform health care units, who are required by contract to visit the sick within three days.

The agent can also make work plans for the future taking into account the wishes of the working force, the reports of the health care units and the requirements from the employer. The advantage of such an agent is that processes with a high priority can be guaranteed to be available under varying circumstances.

A.4.3.3 Issues with labour resource management and information management.

When the counter assistant agents independently disseminate information to the citizen, what is the position of the agent in case that the information is wrong? Was the citizen (or its representative agent) not eloquent enough in stating its wish, and is it the duty of the citizen to state his question right, or is it the duty of the information agent to resolve any ambiguity? What happens when the agent, at the time of the question, has answered right, but in the term of the procedure, the procedure has been changed? Who is responsible for the repercussions? Also when several parties are responsible in a procedure -as is illustrated in the first situation where the counter

assistant agent has started the production of the passport and the human steps in to finalise the procedure-, who should the citizen talk to when something has gone wrong? Should the citizen keep the counter person responsible or should the counter assistant agent, who started the procedure, be responsible, or should both be responsible or … What should be done?

When the counter assistant agents add to the body of knowledge this could produce conflicting or contradictory information, what measures should be taken to prevent this?

Another issue that can be raised is the increased co-operation between agent and man; they are part of the working condition of people. These interactions could lead to responsibilities who are not well defined, or lead to an increase in efficiency that is not in pace with human ability, how should this be resolved?

What happens when the agent does not schedule correctly people to do other work? Who is then responsible? The policy maker who designed the perimeters of the procedures, the designer of the optimisation algorithms, the builder of the agent software or the town hall who uses the agents?

B LEGAL FRAMEWORK

The main problem that has to be solved is to answer the question what legal requirements the designers of software agents have to take into account when going about their business. This raises the question what these legal requirements are. As a possible answer the identification of a number of abstract roles that a software agent can play is proposed. A software agent will in every role be subject to the legal requirements that 'come with' this role. The roles thus serve the purpose of structuring the description of the legal requirements. In order to keep things simple the choice of roles will be inspired by the legal conceptualization of the world. For example, an agent may play the role of a provider of services. The legal requirements that come with this role may be fairly easy determinable, since the relevant legal regulations take the perspective of the service provider as a starting point; see for instance the directive on e-commerce which regulates the provision of services of the information society. The legal perspective is justified, since the central problem that was put before us is of a legal nature: What are the legal requirements that have to be met when designing a software agent? Furthermore, an important advantage is that the roles can be chosen in such a way that they are from a (legal) layman's perspective very recognizable.

Furthermore, the roles will not be disjunct. For example, an agent used to conclude contracts (i.e., a role) will generally at the same time act as a processor of data (i.e., another role). The assumption of more than one role at any one time is not necessarily a problem. At some point, a division is to be made of sub problems in to sensibly describe the legal aspects of agents. Some roles will be more general than others. A software agent will usually be a processor of data, but only a few agents will be used to conclude contracts.

An advantage of this approach is that the concept is easily extensible with new roles, when the need arises. For example, if governments would decide – and this is purely hypothetical – to make software agents personae in law and introduce a general duty to identify oneself on the Internet, one could introduce the role: the software agent as an actor or perhaps the software agent as a carrier of an identity

The scenarios studied in this project fit in very well with this approach: they can be seen as case studies for differing roles. These case studies bring to light in what way the law regarding the several roles comes into play when designing agents, possibly leading to more generalised roles. The central problem is to discover the legal requirements the builder (or user) of software agents has to take into account, not just for (roles of) software agents that fit the scenario's or traces. In sum, additional roles are taken into account than the roles distinguished and studied in ALIAS scenarios.

B.1 WHAT ROLES OF AGENTS ARE TO BE CHOSEN?

A close examination of the scenarios shows that software agents may play the following roles: the agent as data processor, the agent as a processor of personal data, the agent as a means of communication, the agent as a service provider, the agent as a means to conclude contracts and finally the agent as a product that may be consumed. The following sections clarify these roles.

The agent as a data processor

The processing of data is subject of legal interest in the sense that society is come to rely on electronic data for its day-to-day functioning. Therefore, there are legal rules that protect the confidentiality, integrity and availability - the CIA interests - of data. The criminalisation of hacking could be mentioned as a rule of law that finds its origin in the aforementioned concern. For the context of agents another offence seems to have a larger bearing: the criminalisation of the 'destruction' of data. To be more precise, Art. 350b DPC (Dutch Penal Code) reads – in an unofficial translation – as follows:

> "He to whose carelessness it is to blame that data that have been stored, or are processed or transported by means of a computer, are unlawfully modified, erased, rendered useless or inaccessible, or that other data are added to these data, will, if this amounts to serious damage to the data, be punished with imprisonment for the duration of at most a month or a fine of the second category."

It is generally held that, on the basis this article, there exists a duty to secure data against unauthorized manipulation. Article 350b leaves to a large extent open to whom this duty applies, but in the present context the article may be applicable to system administrators, designers and builders of software agents, those who make

agents available for use by third parties etc. Preventing that software agents manipulate data themselves may not even be the most important concern of these persons. Software agents may often have a role in protecting data against unauthorized manipulations by third parties. The persons mentioned are instrumental

The agent as a processor of personal data

The unconstrained use of personal data may harm the persons about whom they convey information. In order to combat (the) adverse effects personal data may have for the persons concerned, the processing of personal data is regulated by law. In the Netherlands the WBP is the main source of law in this respect. The WBP addresses *inter alia* items such as the purpose for which personal data are collected, the grounds for processing, the duration of storage, etc. The use of software agents often involves the processing of personal data; one only needs to think of fields of application of software agents, such as communication or the conclusion of contracts.

The agent as a means of communication

Communications are in various ways subject to legal regulation. Foremost, there is the telecommunications law, which has as its prime objective to regulate the market for telecommunication networks and services. Telecommunication law regulates, e.g., interoperability between systems, e.g., interconnection (interoperability between networks) or technical specifications of terminal equipment (interoperability between devices and the network).

Furthermore, there is Criminal Procedural Law. The primary goal of this law is to provide a legal basis for and to regulate the access of police to third party communications and the data that are generated by telecommunication services in the process. This may lay duties on providers of telecommunication services, such as the duty to adapt their network in such a way that 'eavesdropping' on the communications over the network is possible, the duty to preserve/store traffic for a certain period etc. In addition, rules about the maximum key length in cryptography belong to this category. Note that the law regarding secret services could be mentioned as well, but is dealt with in this document.

Finally, the law protects the confidentiality of personal communications when using means of telecommunication. To a certain extent this law is the complement of the Criminal Procedural Law mentioned above. The 'telecommunications secret' provides the legal basis for confidentiality of data that are exchanged, the Criminal Procedural Law is the legal basis for infractions on the telecommunications secret by the police (it does so by stating the conditions under which such infraction are allowed for).

The agent as a provider of services

The regulation of the provision of services may serve (one or more of) several purposes, such as consumer protection and fair competition. The directive on e-commerce imposes on providers of information society services the duty to make known their identity and certain other data that consumers have an interest to know, if doing business with the service provider. Commercial communications have to meet certain requirements. It must *inter alia* be clearly indicated from whom they originate and they must be easily identifiable as commercial communications. As software agents are being used for providing services: they will have to conform to the requirements of the directive.

The agent as a means to conclude contracts

Parties may conclude contracts by uttering declarations that express their consent about the subject of the contract. A software agent can be used for these declaratory purposes. Nonetheless, e-contracting and contracting with the help of software agents introduces various kinds of problems: how do I know that the other party is the person whom he pretends to be? How do I prove at later point in time that a contract has been concluded? How do I proof what the contents of the declaration or contract are? How do I prove that an electronic signature is genuine?

The directive on electronic signatures aims to bring about more certainty concerning the evidentiary value of electronic signatures. It does so by various means, such as: discerning between 'normal' electronic signatures and 'advanced' electronic signatures, by laying the basis for voluntary accreditation of certification authorities etc. In the annexes, general requirements for signature technologies, certification bodies and certification procedures are enumerated. In order to translate these general requirements to technical requirements, standardization bodies have bundled their forces in EESSI and are forming technical signature-standards for

various applications, such as default signatures, signatures for low value contracts, signatures that are verifiable after a long period of time. Software agents will in future probably be used to sign documents electronically. What additional problems are being introduced by the fact that it is a software agent that signs?

The agent as a product to be consumed

Apart from the fact that an agent is an entity that acts in a network environment, a software agent is also a product that may or may not meet the expectations of its user. This perspective gives rise to legal questions, such as: from which sources can a user draw (legally respectable) expectations with respect to the abilities of the agent? In what cases is it necessary to provide information, advise or warnings about the functioning of an agent?

B.2 CONCLUDING REMARKS

The above description of the roles needs to be extended via an an iterative process of studying legal documents and discovering new relevant aspects by analyzing scenarios. The aforementioned arrangement of legal problems according to the roles can be used as another format for describing legal aspects of roles of software agents; a possible research endeavour for ALIAS.